Information Retrival

In [5]:

```
!python -m ipykernal install --user --name mykernal
```

```
/Users/krutheekarajkumar/anaconda3/bin/python: No module named ipykernal
```

In [7]:

```
!pip install whoosh
```

```
Collecting whoosh
  Downloading
https://files.pythonhosted.org/packages/ba/19/24d0f1f454a2c1eb689ca28d2f178db81e5024f42d82729a4ff67
5cf/Whoosh-2.7.4-py2.py3-none-any.whl (468kB)
     |████████████████████████████████| 471kB 6.4MB/s eta 0:00:01
Installing collected packages: whoosh
Successfully installed whoosh-2.7.4
```

In [8]:

```
from whoosh import index, writing
from whoosh.fields import Schema, TEXT, KEYWORD, ID, STORED
from whoosh.analysis import *
from whoosh.qparser import QueryParser
import os.path
from pathlib import Path
import tempfile
import subprocess
```

In [9]:

```
DATA_DIR = "government"
DOCUMENTS_DIR = os.path.join(DATA_DIR, "documents")
TOPIC_FILE = os.path.join(DATA_DIR, "gov.topics")
QRELS_FILE = os.path.join(DATA_DIR, "gov.qrels")

#For mac:
TREC_EVAL = os.path.join("trec_eval", "trec_eval")
```

Trec eval's measures that would be appropriate for measuring search system performance for government web sites

The Chosen measure is: **MAP - mean average precision**

Reason for choosing Average Precision

Average Precision is the average of the precision value obtained for the set of top 'k' documents existing after each relevant document is retrieved, therefore this measure considers precision and takes into account the rank.

In [11]:

```
mySchema = Schema(file_path = ID(stored=True),
                  file_content = TEXT(analyzer = RegexTokenizer()))
def createIndex(schema):
    indexDir = tempfile.mkdtemp()
    return index.create_in(indexDir, schema)

myIndex = createIndex(mySchema)
```

In [12]:

```
def addFilesToIndex(indexObj, fileList):
```

```
def addFilesToIndex(indexObj, fileList):
    # open writer
    writer = writing.BufferedWriter(indexObj, period=None, limit=1000)
    #print(fileList)
    #print(indexObj)
    try:
        # write each file to index
        # Here 'fileList' is an object that is iterable - 'lab-data/documents/email01'
        for docNum, filePath in enumerate(fileList):
            with open(filePath, "r", encoding="utf-8") as f:
                fileContent = f.read()
                writer.add_document(file_path = filePath,
                                    file_content = fileContent)
                if (docNum+1 % 1000 == 0):
                    print("already indexed:", docNum+1)
        print("done indexing.")

    finally:
        # close the index
        writer.close()
```

In [13]:

```
filesToIndex = [str(filePath) for filePath in Path(DOCUMENTS_DIR).glob("**/*") if filePath.is_file(
)]
```

In [14]:

```
filesToIndex[:5]
```

Out[14]:

```
['government/documents/61/G00-61-2800209',
 'government/documents/61/G00-61-1192048',
 'government/documents/61/G00-61-1118212',
 'government/documents/61/G00-61-0749882',
 'government/documents/61/G00-61-2230501']
```

In [15]:

```
print("number of files:", len(filesToIndex))
```

```
number of files: 4078
```

In [16]:

```
addFilesToIndex(myIndex, filesToIndex)
```

```
done indexing.
```

In [17]:

```
mySchema = Schema(file_path = ID(stored=True),
                  file_content = TEXT(analyzer = RegexTokenizer()))
INDEX_Q2 = createIndex(mySchema) # Replace None with index
QP_Q2 = QueryParser("file_content", schema=myIndex.schema) # Replace None with query parser
SEARCHER_Q2 = myIndex.searcher()# Replace None with searcher
```

In [18]:

```
with open(TOPIC_FILE, "r") as f:
    print(f.read())
```

```
1 mining gold silver coal
2 juvenile delinquency
4 wireless communications
6 physical therapists
7 cotton industry
9 genealogy searches
10 Physical Fitness
```

```
14 Agricultural biotechnology
16 Emergency and disaster preparedness assistance
18 Shipwrecks
19 Cybercrime, internet fraud, and cyber fraud
22 Veteran's Benefits
24 Air Bag Safety
26 Nuclear power plants
28 Early Childhood Education
```

In [189]:

```
with open(QRELS_FILE, "r") as f:
    qrels10 = f.readlines()[:10]
    print("".join(qrels10))
```

```
1 0 G00-00-0681214 0
1 0 G00-00-0945765 0
1 0 G00-00-1006224 1
1 0 G00-00-1591495 0
1 0 G00-00-2764912 0
1 0 G00-00-3253540 0
1 0 G00-00-3717374 0
1 0 G00-01-0270065 0
1 0 G00-01-0400712 0
1 0 G00-01-0682299 0
```

In [190]:

```
def trecEval(topicFile, qrelsFile, queryParser, searcher):
    # Load topic file - a list of topics(search phrases) used for evalutation
    with open(topicFile, "r") as tf:
        topics = tf.read().splitlines()
    #print(topics) # -- list
    # create an output file to which we'll write our results
    tempOutputFile = tempfile.mkstemp()[1]
    with open(tempOutputFile, "w") as outputTRECFile:
        # for each evaluated topic:
        # build a query and record the results in the file in TREC_EVAL format
        for topic in topics:
            topic_id, topic_phrase = tuple(topic.split(" ", 1))
            topicQuery = queryParser.parse(topic_phrase)
            topicResults = searcher.search(topicQuery, limit=None)
            for (docnum, result) in enumerate(topicResults):
                score = topicResults.score(docnum)
                outputTRECFile.write("%s Q0 %s %d %lf test\n" % (topic_id, os.path.basename(result[
file_path"]), docnum, score))

    result = subprocess.run([TREC_EVAL, '-q', qrelsFile, tempOutputFile], stdout=subprocess.PIPE)
    print(result.stdout.decode())
```

In [191]:

```
trecEval(TOPIC_FILE, QRELS_FILE, QP_Q2, SEARCHER_Q2)
```

```
num_ret         1 3
num_rel         1 5
num_rel_ret     1 0
map             1 0.0000
R-prec          1 0.0000
bpref           1 0.0000
recip_rank      1 0.0000
ircl_prn.0.00   1 0.0000
ircl_prn.0.10   1 0.0000
ircl_prn.0.20   1 0.0000
ircl_prn.0.30   1 0.0000
ircl_prn.0.40   1 0.0000
ircl_prn.0.50   1 0.0000
ircl_prn.0.60   1 0.0000
ircl_prn.0.70   1 0.0000
ircl_prn.0.80   1 0.0000
ircl_prn.0.90   1 0.0000
```

```
ircl_prn.1.00   1 0.0000
P5              1 0.0000
P10             1 0.0000
P15             1 0.0000
P20             1 0.0000
P30             1 0.0000
P100            1 0.0000
P200            1 0.0000
P500            1 0.0000
P1000           1 0.0000
num_ret         2 13
num_rel         2 2
num_rel_ret     2 1
map             2 0.5000
R-prec          2 0.5000
bpref           2 0.5000
recip_rank      2 1.0000
ircl_prn.0.00   2 1.0000
ircl_prn.0.10   2 1.0000
ircl_prn.0.20   2 1.0000
ircl_prn.0.30   2 1.0000
ircl_prn.0.40   2 1.0000
ircl_prn.0.50   2 1.0000
ircl_prn.0.60   2 0.0000
ircl_prn.0.70   2 0.0000
ircl_prn.0.80   2 0.0000
ircl_prn.0.90   2 0.0000
ircl_prn.1.00   2 0.0000
P5              2 0.2000
P10             2 0.1000
P15             2 0.0667
P20             2 0.0500
P30             2 0.0333
P100            2 0.0100
P200            2 0.0050
P500            2 0.0020
P1000           2 0.0010
num_ret         4 40
num_rel         4 4
num_rel_ret     4 3
map             4 0.5357
R-prec          4 0.5000
bpref           4 0.5000
recip_rank      4 1.0000
ircl_prn.0.00   4 1.0000
ircl_prn.0.10   4 1.0000
ircl_prn.0.20   4 1.0000
ircl_prn.0.30   4 1.0000
ircl_prn.0.40   4 1.0000
ircl_prn.0.50   4 1.0000
ircl_prn.0.60   4 0.1429
ircl_prn.0.70   4 0.1429
ircl_prn.0.80   4 0.0000
ircl_prn.0.90   4 0.0000
ircl_prn.1.00   4 0.0000
P5              4 0.4000
P10             4 0.2000
P15             4 0.1333
P20             4 0.1000
P30             4 0.1000
P100            4 0.0300
P200            4 0.0150
P500            4 0.0060
P1000           4 0.0030
num_ret         6 13
num_rel         6 1
num_rel_ret     6 0
map             6 0.0000
R-prec          6 0.0000
bpref           6 0.0000
recip_rank      6 0.0000
ircl_prn.0.00   6 0.0000
ircl_prn.0.10   6 0.0000
ircl_prn.0.20   6 0.0000
ircl_prn.0.30   6 0.0000
ircl_prn.0.40   6 0.0000
ircl_prn.0.50   6 0.0000
```

```
ircl_prn.0.60   6 0.0000
ircl_prn.0.70   6 0.0000
ircl_prn.0.80   6 0.0000
ircl_prn.0.90   6 0.0000
ircl_prn.1.00   6 0.0000
P5              6 0.0000
P10             6 0.0000
P15             6 0.0000
P20             6 0.0000
P30             6 0.0000
P100            6 0.0000
P200            6 0.0000
P500            6 0.0000
P1000           6 0.0000
num_ret         7 14
num_rel         7 3
num_rel_ret     7 0
map             7 0.0000
R-prec          7 0.0000
bpref           7 0.0000
recip_rank      7 0.0000
ircl_prn.0.00   7 0.0000
ircl_prn.0.10   7 0.0000
ircl_prn.0.20   7 0.0000
ircl_prn.0.30   7 0.0000
ircl_prn.0.40   7 0.0000
ircl_prn.0.50   7 0.0000
ircl_prn.0.60   7 0.0000
ircl_prn.0.70   7 0.0000
ircl_prn.0.80   7 0.0000
ircl_prn.0.90   7 0.0000
ircl_prn.1.00   7 0.0000
P5              7 0.0000
P10             7 0.0000
P15             7 0.0000
P20             7 0.0000
P30             7 0.0000
P100            7 0.0000
P200            7 0.0000
P500            7 0.0000
P1000           7 0.0000
num_ret         9 27
num_rel         9 1
num_rel_ret     9 1
map             9 0.0625
R-prec          9 0.0000
bpref           9 0.0000
recip_rank      9 0.0625
ircl_prn.0.00   9 0.0625
ircl_prn.0.10   9 0.0625
ircl_prn.0.20   9 0.0625
ircl_prn.0.30   9 0.0625
ircl_prn.0.40   9 0.0625
ircl_prn.0.50   9 0.0625
ircl_prn.0.60   9 0.0625
ircl_prn.0.70   9 0.0625
ircl_prn.0.80   9 0.0625
ircl_prn.0.90   9 0.0625
ircl_prn.1.00   9 0.0625
P5              9 0.0000
P10             9 0.0000
P15             9 0.0000
P20             9 0.0500
P30             9 0.0333
P100            9 0.0100
P200            9 0.0050
P500            9 0.0020
P1000           9 0.0010
num_ret         10 42
num_rel         10 1
num_rel_ret     10 1
map             10 0.2000
R-prec          10 0.0000
bpref           10 0.0000
recip_rank      10 0.2000
ircl_prn.0.00   10 0.2000
ircl_prn.0.10   10 0.2000
```

```
ircl_prn.0.20   10 0.2000
ircl_prn.0.30   10 0.2000
ircl_prn.0.40   10 0.2000
ircl_prn.0.50   10 0.2000
ircl_prn.0.60   10 0.2000
ircl_prn.0.70   10 0.2000
ircl_prn.0.80   10 0.2000
ircl_prn.0.90   10 0.2000
ircl_prn.1.00   10 0.2000
P5              10 0.2000
P10             10 0.1000
P15             10 0.0667
P20             10 0.0500
P30             10 0.0333
P100            10 0.0100
P200            10 0.0050
P500            10 0.0020
P1000           10 0.0010
num_ret         14 26
num_rel         14 1
num_rel_ret     14 1
map             14 1.0000
R-prec          14 1.0000
bpref           14 1.0000
recip_rank      14 1.0000
ircl_prn.0.00   14 1.0000
ircl_prn.0.10   14 1.0000
ircl_prn.0.20   14 1.0000
ircl_prn.0.30   14 1.0000
ircl_prn.0.40   14 1.0000
ircl_prn.0.50   14 1.0000
ircl_prn.0.60   14 1.0000
ircl_prn.0.70   14 1.0000
ircl_prn.0.80   14 1.0000
ircl_prn.0.90   14 1.0000
ircl_prn.1.00   14 1.0000
P5              14 0.2000
P10             14 0.1000
P15             14 0.0667
P20             14 0.0500
P30             14 0.0333
P100            14 0.0100
P200            14 0.0050
P500            14 0.0020
P1000           14 0.0010
num_ret         16 17
num_rel         16 7
num_rel_ret     16 0
map             16 0.0000
R-prec          16 0.0000
bpref           16 0.0000
recip_rank      16 0.0000
ircl_prn.0.00   16 0.0000
ircl_prn.0.10   16 0.0000
ircl_prn.0.20   16 0.0000
ircl_prn.0.30   16 0.0000
ircl_prn.0.40   16 0.0000
ircl_prn.0.50   16 0.0000
ircl_prn.0.60   16 0.0000
ircl_prn.0.70   16 0.0000
ircl_prn.0.80   16 0.0000
ircl_prn.0.90   16 0.0000
ircl_prn.1.00   16 0.0000
P5              16 0.0000
P10             16 0.0000
P15             16 0.0000
P20             16 0.0000
P30             16 0.0000
P100            16 0.0000
P200            16 0.0000
P500            16 0.0000
P1000           16 0.0000
num_ret         18 32
num_rel         18 1
num_rel_ret     18 1
map             18 1.0000
R-prec          18 1.0000
```

```
R-prec          18 1.0000
bpref           18 1.0000
recip_rank      18 1.0000
ircl_prn.0.00   18 1.0000
ircl_prn.0.10   18 1.0000
ircl_prn.0.20   18 1.0000
ircl_prn.0.30   18 1.0000
ircl_prn.0.40   18 1.0000
ircl_prn.0.50   18 1.0000
ircl_prn.0.60   18 1.0000
ircl_prn.0.70   18 1.0000
ircl_prn.0.80   18 1.0000
ircl_prn.0.90   18 1.0000
ircl_prn.1.00   18 1.0000
P5              18 0.2000
P10             18 0.1000
P15             18 0.0667
P20             18 0.0500
P30             18 0.0333
P100            18 0.0100
P200            18 0.0050
P500            18 0.0020
P1000           18 0.0010
num_ret         19 5
num_rel         19 2
num_rel_ret     19 1
map             19 0.5000
R-prec          19 0.5000
bpref           19 0.5000
recip_rank      19 1.0000
ircl_prn.0.00   19 1.0000
ircl_prn.0.10   19 1.0000
ircl_prn.0.20   19 1.0000
ircl_prn.0.30   19 1.0000
ircl_prn.0.40   19 1.0000
ircl_prn.0.50   19 1.0000
ircl_prn.0.60   19 0.0000
ircl_prn.0.70   19 0.0000
ircl_prn.0.80   19 0.0000
ircl_prn.0.90   19 0.0000
ircl_prn.1.00   19 0.0000
P5              19 0.2000
P10             19 0.1000
P15             19 0.0667
P20             19 0.0500
P30             19 0.0333
P100            19 0.0100
P200            19 0.0050
P500            19 0.0020
P1000           19 0.0010
num_ret         22 107
num_rel         22 1
num_rel_ret     22 1
map             22 0.0435
R-prec          22 0.0000
bpref           22 0.0000
recip_rank      22 0.0435
ircl_prn.0.00   22 0.0435
ircl_prn.0.10   22 0.0435
ircl_prn.0.20   22 0.0435
ircl_prn.0.30   22 0.0435
ircl_prn.0.40   22 0.0435
ircl_prn.0.50   22 0.0435
ircl_prn.0.60   22 0.0435
ircl_prn.0.70   22 0.0435
ircl_prn.0.80   22 0.0435
ircl_prn.0.90   22 0.0435
ircl_prn.1.00   22 0.0435
P5              22 0.0000
P10             22 0.0000
P15             22 0.0000
P20             22 0.0000
P30             22 0.0333
P100            22 0.0100
P200            22 0.0050
P500            22 0.0020
P1000           22 0.0010
num_ret         24 20
```

```
num_ret         24 20
num_rel         24 1
num_rel_ret     24 1
map             24 1.0000
R-prec          24 1.0000
bpref           24 1.0000
recip_rank      24 1.0000
ircl_prn.0.00   24 1.0000
ircl_prn.0.10   24 1.0000
ircl_prn.0.20   24 1.0000
ircl_prn.0.30   24 1.0000
ircl_prn.0.40   24 1.0000
ircl_prn.0.50   24 1.0000
ircl_prn.0.60   24 1.0000
ircl_prn.0.70   24 1.0000
ircl_prn.0.80   24 1.0000
ircl_prn.0.90   24 1.0000
ircl_prn.1.00   24 1.0000
P5              24 0.2000
P10             24 0.1000
P15             24 0.0667
P20             24 0.0500
P30             24 0.0333
P100            24 0.0100
P200            24 0.0050
P500            24 0.0020
P1000           24 0.0010
num_ret         26 51
num_rel         26 3
num_rel_ret     26 2
map             26 0.0771
R-prec          26 0.0000
bpref           26 0.0000
recip_rank      26 0.1667
ircl_prn.0.00   26 0.1667
ircl_prn.0.10   26 0.1667
ircl_prn.0.20   26 0.1667
ircl_prn.0.30   26 0.1667
ircl_prn.0.40   26 0.0645
ircl_prn.0.50   26 0.0645
ircl_prn.0.60   26 0.0645
ircl_prn.0.70   26 0.0000
ircl_prn.0.80   26 0.0000
ircl_prn.0.90   26 0.0000
ircl_prn.1.00   26 0.0000
P5              26 0.0000
P10             26 0.1000
P15             26 0.0667
P20             26 0.0500
P30             26 0.0333
P100            26 0.0200
P200            26 0.0100
P500            26 0.0040
P1000           26 0.0020
num_ret         28 34
num_rel         28 2
num_rel_ret     28 1
map             28 0.0227
R-prec          28 0.0000
bpref           28 0.0000
recip_rank      28 0.0455
ircl_prn.0.00   28 0.0455
ircl_prn.0.10   28 0.0455
ircl_prn.0.20   28 0.0455
ircl_prn.0.30   28 0.0455
ircl_prn.0.40   28 0.0455
ircl_prn.0.50   28 0.0455
ircl_prn.0.60   28 0.0000
ircl_prn.0.70   28 0.0000
ircl_prn.0.80   28 0.0000
ircl_prn.0.90   28 0.0000
ircl_prn.1.00   28 0.0000
P5              28 0.0000
P10             28 0.0000
P15             28 0.0000
P20             28 0.0000
P30             28 0.0333
P100            28 0.0100
```

```
P100            28 0.0100
P200            28 0.0050
P500            28 0.0020
P1000           28 0.0010
num_q           all 15
num_ret         all 444
num_rel         all 35
num_rel_ret     all 14
map             all 0.3294
gm_ap           all 0.0161
R-prec          all 0.3000
bpref           all 0.3000
recip_rank      all 0.4345
ircl_prn.0.00   all 0.4345
ircl_prn.0.10   all 0.4345
ircl_prn.0.20   all 0.4345
ircl_prn.0.30   all 0.4345
ircl_prn.0.40   all 0.4277
ircl_prn.0.50   all 0.4277
ircl_prn.0.60   all 0.2342
ircl_prn.0.70   all 0.2299
ircl_prn.0.80   all 0.2204
ircl_prn.0.90   all 0.2204
ircl_prn.1.00   all 0.2204
P5              all 0.1067
P10             all 0.0600
P15             all 0.0400
P20             all 0.0333
P30             all 0.0289
P100            all 0.0093
P200            all 0.0047
P500            all 0.0019
P1000           all 0.0009
```

Q2 (b): How well did the baseline Whoosh system do on your chosen measure? [Provide the number.]

map(all) = **0.1971**

Q2 (c): Are there any particular topics where it did very well, or very badly? [If so, list a few topic IDs for each]

The topic where the MAP value was 0 were: 1, 2, 6, 7, 9, 16, 28

Topic 19 wasn't included in the list and the reason for this is that there were no documents that were retrived by the system.

The topic which did have a MAP value as a consequent of having a system retriving a relevant document: 4=0.0312 , 10=0.167, 14=0.25, 18=1.0, 22=0.2, 24=1.0, 26=0.1111

Gauging from the returned documents, topic 14 had the most promising results.

Rason for improvement of Whoosh's performance on this test collection

The first way to improve the test collections would be to introduce text tokenization and filteration.
For Example:
The queries need not consider any puncuation or unneccesary white space. Uppercase and lowercase letters need not be a deciding factor in query searching. Using different variations of a query term which has the same meaning, however could be listed different in the context due to grammer but carry the same meaning.

In [104]:

```
import nltk
from nltk.stem import *
```

In [192]:

```
from whoosh.analysis import Filter
class CustomFilter(Filter):
    is_morph = True
    def __init__(self, filterFunc, *args, **kwargs):
```

```
    def __init__(self, filterFunc, *args, **kwargs):
        self.customFilter = filterFunc
        self.args = args
        self.kwargs = kwargs
    def __eq__(self):
        return (other
                and self.__class__ is other.__class__)
    def __call__(self, tokens):
        for t in tokens:
            if t.mode == 'query': # if called by query parser
                t.text = self.customFilter(t.text, *self.args, **self.kwargs)
                yield t
            else: # == 'index' if called by indexer
                t.text = self.customFilter(t.text, *self.args, **self.kwargs)
                yield t
```

In [195]:

```
tokenizer = RegexTokenizer()| StemFilter()| LowercaseFilter() | IntraWordFilter() | StopFilter() |C
ustomFilter(WordNetLemmatizer().lemmatize, 'v')|CustomFilter(LancasterStemmer().stem)
#[token.text for token in stmLwrStpIntraAnalyzer()]
```

In [196]:

```
Schema_Q3 = Schema(file_path = ID(stored=True),
                   file_content = TEXT(analyzer = tokenizer))
INDEX_Q3 = createIndex(Schema_Q3) # Replace None with your index for Q3
addFilesToIndex(INDEX_Q3, filesToIndex)
QP_Q3 =  QueryParser("file_content", schema=INDEX_Q3.schema) # Replace None with your query parser
for Q3
SEARCHER_Q3 = INDEX_Q3.searcher() # Replace None with your searcher for Q3
```

```
done indexing.
```

In [197]:

```
trecEval(TOPIC_FILE, QRELS_FILE, QP_Q3, SEARCHER_Q3)
```

```
num_ret         1 3
num_rel         1 5
num_rel_ret     1 0
map             1 0.0000
R-prec          1 0.0000
bpref           1 0.0000
recip_rank      1 0.0000
ircl_prn.0.00   1 0.0000
ircl_prn.0.10   1 0.0000
ircl_prn.0.20   1 0.0000
ircl_prn.0.30   1 0.0000
ircl_prn.0.40   1 0.0000
ircl_prn.0.50   1 0.0000
ircl_prn.0.60   1 0.0000
ircl_prn.0.70   1 0.0000
ircl_prn.0.80   1 0.0000
ircl_prn.0.90   1 0.0000
ircl_prn.1.00   1 0.0000
P5              1 0.0000
P10             1 0.0000
P15             1 0.0000
P20             1 0.0000
P30             1 0.0000
P100            1 0.0000
P200            1 0.0000
P500            1 0.0000
P1000           1 0.0000
num_ret         2 13
num_rel         2 2
num_rel_ret     2 1
map             2 0.5000
R-prec          2 0.5000
bpref           2 0.5000
recip_rank      2 1.0000
ircl_prn.0.00   2 1.0000
```

```
ircl_prn.0.10   2 1.0000
ircl_prn.0.20   2 1.0000
ircl_prn.0.30   2 1.0000
ircl_prn.0.40   2 1.0000
ircl_prn.0.50   2 1.0000
ircl_prn.0.60   2 0.0000
ircl_prn.0.70   2 0.0000
ircl_prn.0.80   2 0.0000
ircl_prn.0.90   2 0.0000
ircl_prn.1.00   2 0.0000
P5              2 0.2000
P10             2 0.1000
P15             2 0.0667
P20             2 0.0500
P30             2 0.0333
P100            2 0.0100
P200            2 0.0050
P500            2 0.0020
P1000           2 0.0010
num_ret         4 41
num_rel         4 4
num_rel_ret     4 3
map             4 0.5357
R-prec          4 0.5000
bpref           4 0.5000
recip_rank      4 1.0000
ircl_prn.0.00   4 1.0000
ircl_prn.0.10   4 1.0000
ircl_prn.0.20   4 1.0000
ircl_prn.0.30   4 1.0000
ircl_prn.0.40   4 1.0000
ircl_prn.0.50   4 1.0000
ircl_prn.0.60   4 0.1429
ircl_prn.0.70   4 0.1429
ircl_prn.0.80   4 0.0000
ircl_prn.0.90   4 0.0000
ircl_prn.1.00   4 0.0000
P5              4 0.4000
P10             4 0.2000
P15             4 0.1333
P20             4 0.1000
P30             4 0.1000
P100            4 0.0300
P200            4 0.0150
P500            4 0.0060
P1000           4 0.0030
num_ret         6 40
num_rel         6 1
num_rel_ret     6 1
map             6 0.1667
R-prec          6 0.0000
bpref           6 0.0000
recip_rank      6 0.1667
ircl_prn.0.00   6 0.1667
ircl_prn.0.10   6 0.1667
ircl_prn.0.20   6 0.1667
ircl_prn.0.30   6 0.1667
ircl_prn.0.40   6 0.1667
ircl_prn.0.50   6 0.1667
ircl_prn.0.60   6 0.1667
ircl_prn.0.70   6 0.1667
ircl_prn.0.80   6 0.1667
ircl_prn.0.90   6 0.1667
ircl_prn.1.00   6 0.1667
P5              6 0.0000
P10             6 0.1000
P15             6 0.0667
P20             6 0.0500
P30             6 0.0333
P100            6 0.0100
P200            6 0.0050
P500            6 0.0020
P1000           6 0.0010
num_ret         7 14
num_rel         7 3
num_rel_ret     7 0
map             7 0.0000
R-prec          7 0.0000
```

```
R-prec          7 0.0000
bpref           7 0.0000
recip_rank      7 0.0000
ircl_prn.0.00   7 0.0000
ircl_prn.0.10   7 0.0000
ircl_prn.0.20   7 0.0000
ircl_prn.0.30   7 0.0000
ircl_prn.0.40   7 0.0000
ircl_prn.0.50   7 0.0000
ircl_prn.0.60   7 0.0000
ircl_prn.0.70   7 0.0000
ircl_prn.0.80   7 0.0000
ircl_prn.0.90   7 0.0000
ircl_prn.1.00   7 0.0000
P5              7 0.0000
P10             7 0.0000
P15             7 0.0000
P20             7 0.0000
P30             7 0.0000
P100            7 0.0000
P200            7 0.0000
P500            7 0.0000
P1000           7 0.0000
num_ret         9 27
num_rel         9 1
num_rel_ret     9 1
map             9 0.0588
R-prec          9 0.0000
bpref           9 0.0000
recip_rank      9 0.0588
ircl_prn.0.00   9 0.0588
ircl_prn.0.10   9 0.0588
ircl_prn.0.20   9 0.0588
ircl_prn.0.30   9 0.0588
ircl_prn.0.40   9 0.0588
ircl_prn.0.50   9 0.0588
ircl_prn.0.60   9 0.0588
ircl_prn.0.70   9 0.0588
ircl_prn.0.80   9 0.0588
ircl_prn.0.90   9 0.0588
ircl_prn.1.00   9 0.0588
P5              9 0.0000
P10             9 0.0000
P15             9 0.0000
P20             9 0.0500
P30             9 0.0333
P100            9 0.0100
P200            9 0.0050
P500            9 0.0020
P1000           9 0.0010
num_ret         10 52
num_rel         10 1
num_rel_ret     10 1
map             10 0.2500
R-prec          10 0.0000
bpref           10 0.0000
recip_rank      10 0.2500
ircl_prn.0.00   10 0.2500
ircl_prn.0.10   10 0.2500
ircl_prn.0.20   10 0.2500
ircl_prn.0.30   10 0.2500
ircl_prn.0.40   10 0.2500
ircl_prn.0.50   10 0.2500
ircl_prn.0.60   10 0.2500
ircl_prn.0.70   10 0.2500
ircl_prn.0.80   10 0.2500
ircl_prn.0.90   10 0.2500
ircl_prn.1.00   10 0.2500
P5              10 0.2000
P10             10 0.1000
P15             10 0.0667
P20             10 0.0500
P30             10 0.0333
P100            10 0.0100
P200            10 0.0050
P500            10 0.0020
P1000           10 0.0010
```

```
num_ret         14 26
num_rel         14 1
num_rel_ret     14 1
map             14 1.0000
R-prec          14 1.0000
bpref           14 1.0000
recip_rank      14 1.0000
ircl_prn.0.00   14 1.0000
ircl_prn.0.10   14 1.0000
ircl_prn.0.20   14 1.0000
ircl_prn.0.30   14 1.0000
ircl_prn.0.40   14 1.0000
ircl_prn.0.50   14 1.0000
ircl_prn.0.60   14 1.0000
ircl_prn.0.70   14 1.0000
ircl_prn.0.80   14 1.0000
ircl_prn.0.90   14 1.0000
ircl_prn.1.00   14 1.0000
P5              14 0.2000
P10             14 0.1000
P15             14 0.0667
P20             14 0.0500
P30             14 0.0333
P100            14 0.0100
P200            14 0.0050
P500            14 0.0020
P1000           14 0.0010
num_ret         16 33
num_rel         16 7
num_rel_ret     16 0
map             16 0.0000
R-prec          16 0.0000
bpref           16 0.0000
recip_rank      16 0.0000
ircl_prn.0.00   16 0.0000
ircl_prn.0.10   16 0.0000
ircl_prn.0.20   16 0.0000
ircl_prn.0.30   16 0.0000
ircl_prn.0.40   16 0.0000
ircl_prn.0.50   16 0.0000
ircl_prn.0.60   16 0.0000
ircl_prn.0.70   16 0.0000
ircl_prn.0.80   16 0.0000
ircl_prn.0.90   16 0.0000
ircl_prn.1.00   16 0.0000
P5              16 0.0000
P10             16 0.0000
P15             16 0.0000
P20             16 0.0000
P30             16 0.0000
P100            16 0.0000
P200            16 0.0000
P500            16 0.0000
P1000           16 0.0000
num_ret         18 32
num_rel         18 1
num_rel_ret     18 1
map             18 1.0000
R-prec          18 1.0000
bpref           18 1.0000
recip_rank      18 1.0000
ircl_prn.0.00   18 1.0000
ircl_prn.0.10   18 1.0000
ircl_prn.0.20   18 1.0000
ircl_prn.0.30   18 1.0000
ircl_prn.0.40   18 1.0000
ircl_prn.0.50   18 1.0000
ircl_prn.0.60   18 1.0000
ircl_prn.0.70   18 1.0000
ircl_prn.0.80   18 1.0000
ircl_prn.0.90   18 1.0000
ircl_prn.1.00   18 1.0000
P5              18 0.2000
P10             18 0.1000
P15             18 0.0667
P20             18 0.0500
P30             18 0.0333
```

```
P100            18 0.0100
P200            18 0.0050
P500            18 0.0020
P1000           18 0.0010
num_ret         19 5
num_rel         19 2
num_rel_ret     19 1
map             19 0.5000
R-prec          19 0.5000
bpref           19 0.5000
recip_rank      19 1.0000
ircl_prn.0.00   19 1.0000
ircl_prn.0.10   19 1.0000
ircl_prn.0.20   19 1.0000
ircl_prn.0.30   19 1.0000
ircl_prn.0.40   19 1.0000
ircl_prn.0.50   19 1.0000
ircl_prn.0.60   19 0.0000
ircl_prn.0.70   19 0.0000
ircl_prn.0.80   19 0.0000
ircl_prn.0.90   19 0.0000
ircl_prn.1.00   19 0.0000
P5              19 0.2000
P10             19 0.1000
P15             19 0.0667
P20             19 0.0500
P30             19 0.0333
P100            19 0.0100
P200            19 0.0050
P500            19 0.0020
P1000           19 0.0010
num_ret         22 107
num_rel         22 1
num_rel_ret     22 1
map             22 0.0417
R-prec          22 0.0000
bpref           22 0.0000
recip_rank      22 0.0417
ircl_prn.0.00   22 0.0417
ircl_prn.0.10   22 0.0417
ircl_prn.0.20   22 0.0417
ircl_prn.0.30   22 0.0417
ircl_prn.0.40   22 0.0417
ircl_prn.0.50   22 0.0417
ircl_prn.0.60   22 0.0417
ircl_prn.0.70   22 0.0417
ircl_prn.0.80   22 0.0417
ircl_prn.0.90   22 0.0417
ircl_prn.1.00   22 0.0417
P5              22 0.0000
P10             22 0.0000
P15             22 0.0000
P20             22 0.0000
P30             22 0.0333
P100            22 0.0100
P200            22 0.0050
P500            22 0.0020
P1000           22 0.0010
num_ret         24 23
num_rel         24 1
num_rel_ret     24 1
map             24 1.0000
R-prec          24 1.0000
bpref           24 1.0000
recip_rank      24 1.0000
ircl_prn.0.00   24 1.0000
ircl_prn.0.10   24 1.0000
ircl_prn.0.20   24 1.0000
ircl_prn.0.30   24 1.0000
ircl_prn.0.40   24 1.0000
ircl_prn.0.50   24 1.0000
ircl_prn.0.60   24 1.0000
ircl_prn.0.70   24 1.0000
ircl_prn.0.80   24 1.0000
ircl_prn.0.90   24 1.0000
ircl_prn.1.00   24 1.0000
P5              24 0.2000
```

```
P10             24 0.1000
P15             24 0.0667
P20             24 0.0500
P30             24 0.0333
P100            24 0.0100
P200            24 0.0050
P500            24 0.0020
P1000           24 0.0010
num_ret         26 52
num_rel         26 3
num_rel_ret     26 2
map             26 0.0771
R-prec          26 0.0000
bpref           26 0.0000
recip_rank      26 0.1667
ircl_prn.0.00   26 0.1667
ircl_prn.0.10   26 0.1667
ircl_prn.0.20   26 0.1667
ircl_prn.0.30   26 0.1667
ircl_prn.0.40   26 0.0645
ircl_prn.0.50   26 0.0645
ircl_prn.0.60   26 0.0645
ircl_prn.0.70   26 0.0000
ircl_prn.0.80   26 0.0000
ircl_prn.0.90   26 0.0000
ircl_prn.1.00   26 0.0000
P5              26 0.0000
P10             26 0.1000
P15             26 0.0667
P20             26 0.0500
P30             26 0.0333
P100            26 0.0200
P200            26 0.0100
P500            26 0.0040
P1000           26 0.0020
num_ret         28 52
num_rel         28 2
num_rel_ret     28 2
map             28 0.2262
R-prec          28 0.0000
bpref           28 0.0000
recip_rank      28 0.1667
ircl_prn.0.00   28 0.2857
ircl_prn.0.10   28 0.2857
ircl_prn.0.20   28 0.2857
ircl_prn.0.30   28 0.2857
ircl_prn.0.40   28 0.2857
ircl_prn.0.50   28 0.2857
ircl_prn.0.60   28 0.2857
ircl_prn.0.70   28 0.2857
ircl_prn.0.80   28 0.2857
ircl_prn.0.90   28 0.2857
ircl_prn.1.00   28 0.2857
P5              28 0.0000
P10             28 0.2000
P15             28 0.1333
P20             28 0.1000
P30             28 0.0667
P100            28 0.0200
P200            28 0.0100
P500            28 0.0040
P1000           28 0.0020
num_q           all 15
num_ret         all 520
num_rel         all 35
num_rel_ret     all 16
map             all 0.3571
gm_ap           all 0.0362
R-prec          all 0.3000
bpref           all 0.3000
recip_rank      all 0.4567
ircl_prn.0.00   all 0.4646
ircl_prn.0.10   all 0.4646
ircl_prn.0.20   all 0.4646
ircl_prn.0.30   all 0.4646
ircl_prn.0.40   all 0.4578
ircl_prn.0.50   all 0.4578
```

```
ircl_prn.0.60   all 0.2673
ircl_prn.0.70   all 0.2630
ircl_prn.0.80   all 0.2535
ircl_prn.0.90   all 0.2535
ircl_prn.1.00   all 0.2535
P5              all 0.1067
P10             all 0.0800
P15             all 0.0533
P20             all 0.0433
P30             all 0.0333
P100            all 0.0107
P200            all 0.0053
P500            all 0.0021
P1000           all 0.0011
```

Modifications and improvements made and the effect on queries in performance

There were eight filters and tolkenziers that were applied to the system. These included StemFilter which was able to get multiple versions of the same root word, and stopfilter which was able to ignore the common words and give importance to the rare words by a list of predefined words. IntraWordFilter were able to concatinate hyphenated word, this filter might not have been necessary as the query words did not contain any such hyphenated term. ordNetLemmatizer().lemmatize, 'v, tokenization features creates past tense versions of verbs and LancasterStemmer().stem is able to remove the suffix so as to return the root word for a search funciton It would apprear as though the number of false positives had increased after the return of the filters were applied, there was a larger amount of data that was being retrieved but was not relevant. False positive numbers however, seemed to be held constant, from the previous system.
Overall there was definite improvement from the previous test to the next, the number of queries processed had increased by one, the true positive values had also increased which meant more relevant information was retrieved.The number of False negatives had also increased dramatically.

num_q	all	14	num_q	all	15
num_ret	all	151	num_ret	all	532
num_rel	all	33	num_rel	all	35
num_rel_ret	all	7	num_rel_ret	all	16
map	all	0.1971	map	all	0.3522

.

The above picture compares the data from all the results from the first test to the second.

Final thoughts on the changed:

The idea still indicated an overall improvement in the system. The MAP values over all the queries was higher than that without of the filters and tokenizers. Compromize was seen at only a small portion of the results and this was due to the large number of retrieved documents - however the higher MAP values suggest that the average reader would find their answers within the top few searches.

In [198]:

```
GRAD_STUDENT = True # Changed to true for graduate student deliverable
```

Ways to improve Whoosh's performance on this test collection:

Any user would be more likely to enjoy their search experience when the information/relevant documents are present in an orderly fashion, such that the most relevant topics would be present at the top of the list of the retrived documents. This can be done by using scoring methods provided by whoosh. The position of the ranks of the relevant documents can be read through the "recip rank" number of the trec_eval return. The reciprocal of this number gives the position of the first relevant document present in the list. For Query 6, this document is present in the 6th position, ideally the search system would present this relevant document in the first position as there is only one relevant document in the whole list of retrived document. Scoring methods would allow for this to happen. The default method the searcher applies is the BM15F method and it would be beneficial to study which is there are other cases there a different sorting system would be more effective

In [199]:

```
from whoosh import scoring
```

In [205]:

In [205]:

```
#Schema_Q4 = Schema(file_path = ID(stored=True),file_content = TEXT(analyzer = tokenizer))
INDEX_Q4 = createIndex(Schema_Q3)
addFilesToIndex(INDEX_Q4, filesToIndex)
QP_Q4 =  QueryParser("file_content", schema=INDEX_Q4.schema)
```

```
done indexing.
```

In [216]:

```
#SEARCHER_Q4 = INDEX_Q4.searcher(weighting=scoring.BM25F())
#The following scoring function uses the position of the first occurance of a term in each documen
t to calculate the
#score, so documents with the given term earlier in the document will score higher:
def pos_score_fn(searcher, fieldname, text, matcher):
    poses = matcher.value_as("positions")
    return 1.0 / (poses[0] + 1)
pos_weighting = scoring.FunctionWeighting(pos_score_fn)
mw = scoring.MultiWeighting(scoring.BM25F(), id=scoring.Frequency(), keys=scoring.TF_IDF())
#SEARCHER_Q4 = INDEX_Q4.searcher(weighting=pos_weighting)
#SEARCHER_Q4 = INDEX_Q4.searcher(weighting=scoring.Frequency())
#SEARCHER_Q4 = INDEX_Q4.searcher(weighting=scoring.TF_IDF())
SEARCHER_Q4 = INDEX_Q4.searcher(weighting=mw)

trecEval(TOPIC_FILE, QRELS_FILE, QP_Q4, SEARCHER_Q4)
```

```
num_ret         1 3
num_rel         1 5
num_rel_ret     1 0
map             1 0.0000
R-prec          1 0.0000
bpref           1 0.0000
recip_rank      1 0.0000
ircl_prn.0.00   1 0.0000
ircl_prn.0.10   1 0.0000
ircl_prn.0.20   1 0.0000
ircl_prn.0.30   1 0.0000
ircl_prn.0.40   1 0.0000
ircl_prn.0.50   1 0.0000
ircl_prn.0.60   1 0.0000
ircl_prn.0.70   1 0.0000
ircl_prn.0.80   1 0.0000
ircl_prn.0.90   1 0.0000
ircl_prn.1.00   1 0.0000
P5              1 0.0000
P10             1 0.0000
P15             1 0.0000
P20             1 0.0000
P30             1 0.0000
P100            1 0.0000
P200            1 0.0000
P500            1 0.0000
P1000           1 0.0000
num_ret         2 13
num_rel         2 2
num_rel_ret     2 1
map             2 0.5000
R-prec          2 0.5000
bpref           2 0.5000
recip_rank      2 1.0000
ircl_prn.0.00   2 1.0000
ircl_prn.0.10   2 1.0000
ircl_prn.0.20   2 1.0000
ircl_prn.0.30   2 1.0000
ircl_prn.0.40   2 1.0000
ircl_prn.0.50   2 1.0000
ircl_prn.0.60   2 0.0000
ircl_prn.0.70   2 0.0000
ircl_prn.0.80   2 0.0000
ircl_prn.0.90   2 0.0000
ircl_prn.1.00   2 0.0000
P5              2 0.2000
P10             2 0.1000
P15             2 0.0667
P20             2 0.0500
```

```
P20             2 0.0500
P30             2 0.0333
P100            2 0.0100
P200            2 0.0050
P500            2 0.0020
P1000           2 0.0010
num_ret         4 41
num_rel         4 4
num_rel_ret     4 3
map             4 0.5357
R-prec          4 0.5000
bpref           4 0.5000
recip_rank      4 1.0000
ircl_prn.0.00   4 1.0000
ircl_prn.0.10   4 1.0000
ircl_prn.0.20   4 1.0000
ircl_prn.0.30   4 1.0000
ircl_prn.0.40   4 1.0000
ircl_prn.0.50   4 1.0000
ircl_prn.0.60   4 0.1429
ircl_prn.0.70   4 0.1429
ircl_prn.0.80   4 0.0000
ircl_prn.0.90   4 0.0000
ircl_prn.1.00   4 0.0000
P5              4 0.4000
P10             4 0.2000
P15             4 0.1333
P20             4 0.1000
P30             4 0.1000
P100            4 0.0300
P200            4 0.0150
P500            4 0.0060
P1000           4 0.0030
num_ret         6 40
num_rel         6 1
num_rel_ret     6 1
map             6 0.1667
R-prec          6 0.0000
bpref           6 0.0000
recip_rank      6 0.1667
ircl_prn.0.00   6 0.1667
ircl_prn.0.10   6 0.1667
ircl_prn.0.20   6 0.1667
ircl_prn.0.30   6 0.1667
ircl_prn.0.40   6 0.1667
ircl_prn.0.50   6 0.1667
ircl_prn.0.60   6 0.1667
ircl_prn.0.70   6 0.1667
ircl_prn.0.80   6 0.1667
ircl_prn.0.90   6 0.1667
ircl_prn.1.00   6 0.1667
P5              6 0.0000
P10             6 0.1000
P15             6 0.0667
P20             6 0.0500
P30             6 0.0333
P100            6 0.0100
P200            6 0.0050
P500            6 0.0020
P1000           6 0.0010
num_ret         7 14
num_rel         7 3
num_rel_ret     7 0
map             7 0.0000
R-prec          7 0.0000
bpref           7 0.0000
recip_rank      7 0.0000
ircl_prn.0.00   7 0.0000
ircl_prn.0.10   7 0.0000
ircl_prn.0.20   7 0.0000
ircl_prn.0.30   7 0.0000
ircl_prn.0.40   7 0.0000
ircl_prn.0.50   7 0.0000
ircl_prn.0.60   7 0.0000
ircl_prn.0.70   7 0.0000
ircl_prn.0.80   7 0.0000
ircl_prn.0.90   7 0.0000
ircl_prn.1.00   7 0.0000
```

```
ircl_prn.1.00   7 0.0000
P5              7 0.0000
P10             7 0.0000
P15             7 0.0000
P20             7 0.0000
P30             7 0.0000
P100            7 0.0000
P200            7 0.0000
P500            7 0.0000
P1000           7 0.0000
num_ret         9 27
num_rel         9 1
num_rel_ret     9 1
map             9 0.0588
R-prec          9 0.0000
bpref           9 0.0000
recip_rank      9 0.0588
ircl_prn.0.00   9 0.0588
ircl_prn.0.10   9 0.0588
ircl_prn.0.20   9 0.0588
ircl_prn.0.30   9 0.0588
ircl_prn.0.40   9 0.0588
ircl_prn.0.50   9 0.0588
ircl_prn.0.60   9 0.0588
ircl_prn.0.70   9 0.0588
ircl_prn.0.80   9 0.0588
ircl_prn.0.90   9 0.0588
ircl_prn.1.00   9 0.0588
P5              9 0.0000
P10             9 0.0000
P15             9 0.0000
P20             9 0.0500
P30             9 0.0333
P100            9 0.0100
P200            9 0.0050
P500            9 0.0020
P1000           9 0.0010
num_ret         10 52
num_rel         10 1
num_rel_ret     10 1
map             10 0.2500
R-prec          10 0.0000
bpref           10 0.0000
recip_rank      10 0.2500
ircl_prn.0.00   10 0.2500
ircl_prn.0.10   10 0.2500
ircl_prn.0.20   10 0.2500
ircl_prn.0.30   10 0.2500
ircl_prn.0.40   10 0.2500
ircl_prn.0.50   10 0.2500
ircl_prn.0.60   10 0.2500
ircl_prn.0.70   10 0.2500
ircl_prn.0.80   10 0.2500
ircl_prn.0.90   10 0.2500
ircl_prn.1.00   10 0.2500
P5              10 0.2000
P10             10 0.1000
P15             10 0.0667
P20             10 0.0500
P30             10 0.0333
P100            10 0.0100
P200            10 0.0050
P500            10 0.0020
P1000           10 0.0010
num_ret         14 26
num_rel         14 1
num_rel_ret     14 1
map             14 1.0000
R-prec          14 1.0000
bpref           14 1.0000
recip_rank      14 1.0000
ircl_prn.0.00   14 1.0000
ircl_prn.0.10   14 1.0000
ircl_prn.0.20   14 1.0000
ircl_prn.0.30   14 1.0000
ircl_prn.0.40   14 1.0000
ircl_prn.0.50   14 1.0000
ircl_prn.0.60   14 1.0000
```

```
ircl_prn.0.60   14 1.0000
ircl_prn.0.70   14 1.0000
ircl_prn.0.80   14 1.0000
ircl_prn.0.90   14 1.0000
ircl_prn.1.00   14 1.0000
P5              14 0.2000
P10             14 0.1000
P15             14 0.0667
P20             14 0.0500
P30             14 0.0333
P100            14 0.0100
P200            14 0.0050
P500            14 0.0020
P1000           14 0.0010
num_ret         16 33
num_rel         16 7
num_rel_ret     16 0
map             16 0.0000
R-prec          16 0.0000
bpref           16 0.0000
recip_rank      16 0.0000
ircl_prn.0.00   16 0.0000
ircl_prn.0.10   16 0.0000
ircl_prn.0.20   16 0.0000
ircl_prn.0.30   16 0.0000
ircl_prn.0.40   16 0.0000
ircl_prn.0.50   16 0.0000
ircl_prn.0.60   16 0.0000
ircl_prn.0.70   16 0.0000
ircl_prn.0.80   16 0.0000
ircl_prn.0.90   16 0.0000
ircl_prn.1.00   16 0.0000
P5              16 0.0000
P10             16 0.0000
P15             16 0.0000
P20             16 0.0000
P30             16 0.0000
P100            16 0.0000
P200            16 0.0000
P500            16 0.0000
P1000           16 0.0000
num_ret         18 32
num_rel         18 1
num_rel_ret     18 1
map             18 1.0000
R-prec          18 1.0000
bpref           18 1.0000
recip_rank      18 1.0000
ircl_prn.0.00   18 1.0000
ircl_prn.0.10   18 1.0000
ircl_prn.0.20   18 1.0000
ircl_prn.0.30   18 1.0000
ircl_prn.0.40   18 1.0000
ircl_prn.0.50   18 1.0000
ircl_prn.0.60   18 1.0000
ircl_prn.0.70   18 1.0000
ircl_prn.0.80   18 1.0000
ircl_prn.0.90   18 1.0000
ircl_prn.1.00   18 1.0000
P5              18 0.2000
P10             18 0.1000
P15             18 0.0667
P20             18 0.0500
P30             18 0.0333
P100            18 0.0100
P200            18 0.0050
P500            18 0.0020
P1000           18 0.0010
num_ret         19 5
num_rel         19 2
num_rel_ret     19 1
map             19 0.5000
R-prec          19 0.5000
bpref           19 0.5000
recip_rank      19 1.0000
ircl_prn.0.00   19 1.0000
ircl_prn.0.10   19 1.0000
ircl_prn.0.20   19 1.0000
```

```
ircl_prn.0.20   19 1.0000
ircl_prn.0.30   19 1.0000
ircl_prn.0.40   19 1.0000
ircl_prn.0.50   19 1.0000
ircl_prn.0.60   19 0.0000
ircl_prn.0.70   19 0.0000
ircl_prn.0.80   19 0.0000
ircl_prn.0.90   19 0.0000
ircl_prn.1.00   19 0.0000
P5              19 0.2000
P10             19 0.1000
P15             19 0.0667
P20             19 0.0500
P30             19 0.0333
P100            19 0.0100
P200            19 0.0050
P500            19 0.0020
P1000           19 0.0010
num_ret         22 107
num_rel         22 1
num_rel_ret     22 1
map             22 0.0417
R-prec          22 0.0000
bpref           22 0.0000
recip_rank      22 0.0417
ircl_prn.0.00   22 0.0417
ircl_prn.0.10   22 0.0417
ircl_prn.0.20   22 0.0417
ircl_prn.0.30   22 0.0417
ircl_prn.0.40   22 0.0417
ircl_prn.0.50   22 0.0417
ircl_prn.0.60   22 0.0417
ircl_prn.0.70   22 0.0417
ircl_prn.0.80   22 0.0417
ircl_prn.0.90   22 0.0417
ircl_prn.1.00   22 0.0417
P5              22 0.0000
P10             22 0.0000
P15             22 0.0000
P20             22 0.0000
P30             22 0.0333
P100            22 0.0100
P200            22 0.0050
P500            22 0.0020
P1000           22 0.0010
num_ret         24 23
num_rel         24 1
num_rel_ret     24 1
map             24 1.0000
R-prec          24 1.0000
bpref           24 1.0000
recip_rank      24 1.0000
ircl_prn.0.00   24 1.0000
ircl_prn.0.10   24 1.0000
ircl_prn.0.20   24 1.0000
ircl_prn.0.30   24 1.0000
ircl_prn.0.40   24 1.0000
ircl_prn.0.50   24 1.0000
ircl_prn.0.60   24 1.0000
ircl_prn.0.70   24 1.0000
ircl_prn.0.80   24 1.0000
ircl_prn.0.90   24 1.0000
ircl_prn.1.00   24 1.0000
P5              24 0.2000
P10             24 0.1000
P15             24 0.0667
P20             24 0.0500
P30             24 0.0333
P100            24 0.0100
P200            24 0.0050
P500            24 0.0020
P1000           24 0.0010
num_ret         26 52
num_rel         26 3
num_rel_ret     26 2
map             26 0.0771
R-prec          26 0.0000
bpref           26 0.0000
```

```
bpref           26 0.0000
recip_rank      26 0.1667
ircl_prn.0.00   26 0.1667
ircl_prn.0.10   26 0.1667
ircl_prn.0.20   26 0.1667
ircl_prn.0.30   26 0.1667
ircl_prn.0.40   26 0.0645
ircl_prn.0.50   26 0.0645
ircl_prn.0.60   26 0.0645
ircl_prn.0.70   26 0.0000
ircl_prn.0.80   26 0.0000
ircl_prn.0.90   26 0.0000
ircl_prn.1.00   26 0.0000
P5              26 0.0000
P10             26 0.1000
P15             26 0.0667
P20             26 0.0500
P30             26 0.0333
P100            26 0.0200
P200            26 0.0100
P500            26 0.0040
P1000           26 0.0020
num_ret         28 52
num_rel         28 2
num_rel_ret     28 2
map             28 0.2262
R-prec          28 0.0000
bpref           28 0.0000
recip_rank      28 0.1667
ircl_prn.0.00   28 0.2857
ircl_prn.0.10   28 0.2857
ircl_prn.0.20   28 0.2857
ircl_prn.0.30   28 0.2857
ircl_prn.0.40   28 0.2857
ircl_prn.0.50   28 0.2857
ircl_prn.0.60   28 0.2857
ircl_prn.0.70   28 0.2857
ircl_prn.0.80   28 0.2857
ircl_prn.0.90   28 0.2857
ircl_prn.1.00   28 0.2857
P5              28 0.0000
P10             28 0.2000
P15             28 0.1333
P20             28 0.1000
P30             28 0.0667
P100            28 0.0200
P200            28 0.0100
P500            28 0.0040
P1000           28 0.0020
num_q           all 15
num_ret         all 520
num_rel         all 35
num_rel_ret     all 16
map             all 0.3571
gm_ap           all 0.0362
R-prec          all 0.3000
bpref           all 0.3000
recip_rank      all 0.4567
ircl_prn.0.00   all 0.4646
ircl_prn.0.10   all 0.4646
ircl_prn.0.20   all 0.4646
ircl_prn.0.30   all 0.4646
ircl_prn.0.40   all 0.4578
ircl_prn.0.50   all 0.4578
ircl_prn.0.60   all 0.2673
ircl_prn.0.70   all 0.2630
ircl_prn.0.80   all 0.2535
ircl_prn.0.90   all 0.2535
ircl_prn.1.00   all 0.2535
P5              all 0.1067
P10             all 0.0800
P15             all 0.0533
P20             all 0.0433
P30             all 0.0333
P100            all 0.0107
P200            all 0.0053
P500            all 0.0021
```

```
P1000           all 0.0011
```

Modications that caused improvements

Multiple scoring methods were applied to study the effect in the resulting data that was retrived. Namely

- BM25F(): which confirmed that it was the default method that was applied to the previous dataset
- The function pos_score_fn() [taken from the whoosh documentation]: where the higher the position of the relevant document, the higher the allotted score was for that query
- MultiWeighting() [From whoosh documentation]: Where the primary method was BM25F, and the ID fields had Frequency and the key fields had the TF_ID scoring attached to it.
- Frequency: Where the more number of times a query word is present in the document the higher the score would be
- TF_ID: scored the query based on the term in the document and the document in the collection.

Most of the methods did not show a large improvement on the MAP number and this was because the default method used BM25F, returned the optimal solution. The maximum MAP value for all the topics was seen for position scoring function. The only topics that an improvement was seen were 6, 9, 10, 28. For example, query 6 originally placed the first relevant document in position 6 and after the scoring method was applied, that position was moved to the 1st position. The scoring function places the relevant documents higher up on the list, there by causing the MAP values to also increase.

num_q	all	14	num_q	all	15	num_q	all	15	num_q	all	15
num_ret	all	151	num_ret	all	532	num_ret	all	520	num_ret	all	520
num_rel	all	33	num_rel	all	35	num_rel	all	35	num_rel	all	35
num_rel_ret	all	7	num_rel_ret	all	16	num_rel_ret	all	16	num_rel_ret	all	16
map	all	0.1971	map	all	0.3522	map	all	0.3740	map	all	0.3571

The columns above follow: results without any filters or tokenizations, results with default scoring(BM25F), filters and tokenization, results with the scoring function and results with multiple weights (left to right)

The most improvement was seen for query 28 as the number of retrived documents was reduced from 64 to 52,52 (BM25F, position scoring function, multiweighting respectively), this caused the ranking for the relevant retrivals (true positives) to be placed higher in rank, thereby increasing the MAP value. Almost all queries had a larger MAP value when scored by the default or the multiweight method for induvidual queries, there by indicating that BM25F had the most effective scoring method for this data set.

Final thoughts on the improvements

It was a good idea to realize that the default scoring method BM25F was the best scoring method available for most fields. However there are some fields which would be more effectively sorted by using frequency or TF_IDF, it would be benificial to investigate which particular fields would benefit more from such sorting

Validation

In [217]:

```
# Run the following cells to make sure your code returns the correct value types
```

In [218]:

```
from whoosh.index import FileIndex
from whoosh.qparser import QueryParser
from whoosh.searching import Searcher
import os.path
```

In [219]:

```
assert(isinstance(INDEX_Q2, FileIndex)), "Index Type"
assert(isinstance(QP_Q2, QueryParser)), "Query Parser Type"
assert(isinstance(SEARCHER_Q2, Searcher)), "Searcher Type"
print("Q2 Types Validated")
```

```
Q2 Types Validated
```

Q2 Validation

In [220]:

```
assert(isinstance(INDEX_Q3, FileIndex)), "Index Type"
assert(isinstance(QP_Q3, QueryParser)), "Query Parser Type"
assert(isinstance(SEARCHER_Q3, Searcher)), "Searcher Type"
print("Q3 Types Validated")
```

```
Q3 Types Validated
```

Q3 Validation

In [221]:

```
assert((not GRAD_STUDENT) or isinstance(INDEX_Q4, FileIndex)), "Index Type"
assert((not GRAD_STUDENT) or isinstance(QP_Q4, QueryParser)), "Query Parser Type"
assert((not GRAD_STUDENT) or isinstance(SEARCHER_Q4, Searcher)), "Searcher Type"
print("Q4 Types Validated")
```

```
Q4 Types Validated
```

Q4 Validation (Graduate Students)

WORKS REFERENCED:

https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-ranked-retrieval-results-1.html
https://nlp.stanford.edu/IR-book/pdf/08eval.pdf
https://en.wikipedia.org/wiki/Evaluation_measures_(information_retrieval)
https://github.com/usnistgov/trec_eval/tree/master/test
https://media.readthedocs.org/pdf/whoosh/latest/whoosh.pdf

In [ ]:

```

```

In [ ]:

```

```