



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Programme	:	B.Tech	Semester	:	Win Sem 21-22
Course	:	Web Mining	Code	:	CSE3024
Faculty	:	Dr.Bhuvaneswari A	Slot	:	L7+L8
Date	:	25-01-2022	Marks	:	10 Points

Register Number : 19BCE1712

Name : Tanay Vaishnav

Exercise 3: Inverted Index Creation and Searching

Q1 Build the inverted index for the following documents:

ID1: Selenium is a portable framework for testing web applications

ID2: BeautifulSoup is useful for web scraping

ID3: It is a python package for parsing the pages

ID4: Java programming can be used for web applications

ID5: scraping web and crawling web is useful

Procedure:

1. Create .txt files according to the question.
2. We will then pre-process the documents, and then split the documents into tokens.
3. The pre-processing includes conversion to lower case, removal of numbers and other special characters, and stop word removal.
4. Store the pre-processed data in a variable data.
5. After this we need to calculate the number of occurrences of all the words.
6. Also, the position of each word is shown in (x,y) format x being the document number and y being the offset position in that particular document.

Code:

```
import re
documents = ['doc1', 'doc2', 'doc3', 'doc4', 'doc5']
index={}
for id,doc in enumerate(documents):
    filename = doc+".txt"
    with open(filename,'r') as fp:
        data = "".join(fp.readlines())
        data = data.lower()
        ext_words = re.findall(r"([a-z0-9-]+)",data)
        for pos,word in enumerate(ext_words):
            if word[-1]=='s':
                if word[:-1] in index:
                    word = word[:-1]
                elif word[:-2] in index:
                    word = word[:-2]

            if word not in index:
                index[word]={ "freq":1, "listing": [(id+1,pos)] }
            else:
                index[word]['freq']+=1
                index[word]['listing'].append((id+1,pos))
from collections import OrderedDict
index = OrderedDict(sorted(index.items()))
with open("inverted.txt",'w') as fp:
    for key in index:
        print(f"{key} : {index[key]}")
        fp.write(f"{key} : {index[key]}\n")
```

Output:

```
➤ a : {'freq': 2, 'listing': [(1, 2), (3, 2)]}
and : {'freq': 1, 'listing': [(5, 2)]}
applications : {'freq': 2, 'listing': [(1, 8), (4, 7)]}
be : {'freq': 1, 'listing': [(4, 3)]}
beautiful : {'freq': 1, 'listing': [(2, 0)]}
can : {'freq': 1, 'listing': [(4, 2)]}
crawling : {'freq': 1, 'listing': [(5, 3)]}
for : {'freq': 4, 'listing': [(1, 5), (2, 4), (3, 5), (4, 5)]}
framework : {'freq': 1, 'listing': [(1, 4)]}
is : {'freq': 4, 'listing': [(1, 1), (2, 2), (3, 1), (5, 5)]}
it : {'freq': 1, 'listing': [(3, 0)]}
java : {'freq': 1, 'listing': [(4, 0)]}
package : {'freq': 1, 'listing': [(3, 4)]}
pages : {'freq': 1, 'listing': [(3, 8)]}
parsing : {'freq': 1, 'listing': [(3, 6)]}
portable : {'freq': 1, 'listing': [(1, 3)]}
programming : {'freq': 1, 'listing': [(4, 1)]}
python : {'freq': 1, 'listing': [(3, 3)]}
scrapping : {'freq': 2, 'listing': [(2, 6), (5, 0)]}
selenium : {'freq': 1, 'listing': [(1, 0)]}
soup : {'freq': 1, 'listing': [(2, 1)]}
testing : {'freq': 1, 'listing': [(1, 6)]}
the : {'freq': 1, 'listing': [(3, 7)]}
used : {'freq': 1, 'listing': [(4, 4)]}
useful : {'freq': 2, 'listing': [(2, 3), (5, 6)]}
web : {'freq': 5, 'listing': [(1, 7), (2, 5), (4, 6), (5, 1), (5, 4)]}
```

Q2 Search following words using the inverted index

Procedure:

1. Create .txt files according to the question.
2. We will then pre-process the documents, and then split the documents into tokens.
3. The pre-processing includes conversion to lower case, removal of numbers and other special characters, and stop word removal.
4. Store the pre-processed data in a variable data.
5. After this we need to calculate the number of occurrences of the specific word given in the question.
6. The word is searched through regular expression and if the word/s are found then their frequency and position is displayed.
7. The position of the words is shown in (x,y) format x being the document number and y being the offset position in that particular document.

a. Selenium AND web

Code:

```
import re
documents = ['doc1', 'doc2', 'doc3', 'doc4', 'doc5']
index={}
for id,doc in enumerate(documents):
    filename = doc+".txt"
    with open(filename,'r') as fp:
        data = "".join(fp.readlines())
        data = data.lower()
        #print(len(data))
        if re.findall(r"\bselenium\b", data) and re.findall(r"\bweb\b", data):
            print("Match found in", filename)
            ext_words1 = re.findall(r"\bselenium\b", data)
            ext_words2 = re.findall(r"\bweb\b", data)
            for pos,word in enumerate(ext_words1):
                if word not in index:
                    index[word]={ "freq":1, "listing": [(id+1,pos)] }
                else:
                    index[word][ 'freq' ]+=1
                    index[word][ 'listing' ].append((id+1,pos))
            for pos,word in enumerate(ext_words2):
                if word not in index:
                    index[word]={ "freq":1, "listing": [(id+1,pos)] }
                else:
                    index[word][ 'freq' ]+=1
                    index[word][ 'listing' ].append((id+1,pos))
            else:
                print("No match found in", filename)
                #break
print("\n")
from collections import OrderedDict
```

```

index = OrderedDict(sorted(index.items()))
with open("inverted.txt",'w') as fp:
    print("Answer: \n")
    for key in index:
        print(f"{key} : {index[key]}")
        fp.write(f"{key} : {index[key]}\n")

```

Output:

```

➤ Match found in doc1.txt
No match found in doc2.txt
No match found in doc3.txt
No match found in doc4.txt
No match found in doc5.txt

Answer:

selenium : {'freq': 1, 'listing': [(1, 0)]}
web : {'freq': 1, 'listing': [(1, 0)]}

```

b. Soup

Code:

```

import re
documents = ['doc1', 'doc2', 'doc3', 'doc4', 'doc5']
index={}
for id,doc in enumerate(documents):
    filename = doc+".txt"
    with open(filename,'r') as fp:
        data = "".join(fp.readlines())
        data = data.lower()
        #print(len(data))
        ext_words = re.findall(r"\bsoup\b", data)
        for pos,word in enumerate(ext_words):
            if word not in index:
                index[word]={ "freq":1, "listing": [(id+1,pos)] }
            else:
                index[word]['freq']+=1
                index[word]['listing'].append((id+1,pos))
from collections import OrderedDict
index = OrderedDict(sorted(index.items()))
with open("inverted.txt",'w') as fp:
    for key in index:
        print(f"{key} : {index[key]}")
        fp.write(f"{key} : {index[key]}\n")

```

Output:

```

➤ soup : {'freq': 1, 'listing': [(2, 0)]}

```

c. Python OR java

Code:

```
import re
documents = ['doc1', 'doc2', 'doc3', 'doc4', 'doc5']
index={}
for id,doc in enumerate(documents):
    filename = doc+".txt"
    with open(filename,'r') as fp:
        data = "".join(fp.readlines())
        data = data.lower()
        #print(len(data))
    ext_words1 = re.compile(r"\bpython\b | \bjava\b",flags=re.I | re.X)
    ext_words2=ext_words1.findall(data)
    for pos,word in enumerate(ext_words2):
        if word not in index:
            index[word]={ "freq":1, "listing": [(id+1,pos)] }
        else:
            index[word]['freq']+=1
            index[word]['listing'].append((id+1,pos))
from collections import OrderedDict
index = OrderedDict(sorted(index.items()))
with open("inverted.txt",'w') as fp:
    for key in index:
        print(f"{key} : {index[key]}")
        fp.write(f"{key} : {index[key]}\n")
```

Output:

```
java : {'freq': 1, 'listing': [(4, 0)]}
python : {'freq': 1, 'listing': [(3, 0)]}
```

d. Web AND craw

Code:

```
import re
documents = ['doc1', 'doc2', 'doc3', 'doc4', 'doc5']
index={}
for id,doc in enumerate(documents):
    filename = doc+".txt"
    with open(filename,'r') as fp:
        data = "".join(fp.readlines())
        data = data.lower()
        #print(len(data))
    if re.findall(r"\bweb\b", data) and re.findall(r"\bcraw\b", data):
        print("Match found in", filename)
    ext_words1 = re.findall(r"\bweb\b", data)
```

```

ext_words2 = re.findall(r"\bcraw\b", data)
for pos,word in enumerate(ext_words1):
    if word not in index:
        index[word]={"freq":1, "listing": [(id+1,pos)] }
    else:
        index[word]['freq']+=1
        index[word]['listing'].append((id+1,pos))
for pos,word in enumerate(ext_words2):
    if word not in index:
        index[word]={"freq":1, "listing": [(id+1,pos)] }
    else:
        index[word]['freq']+=1
        index[word]['listing'].append((id+1,pos))
else:
    print("No match found in", filename)
    #break
print("\n")
from collections import OrderedDict
index = OrderedDict(sorted(index.items()))
with open("inverted.txt",'w') as fp:
    #print("Answer: \n")
    for key in index:
        print(f"{key} : {index[key]}")
        fp.write(f"{key} : {index[key]}\n")

```

Output:

```

➡ No match found in doc1.txt
No match found in doc2.txt
No match found in doc3.txt
No match found in doc4.txt
No match found in doc5.txt

```

CODE FILE:



QUESTION 1

```
[ ] import re
documents = ['doc1', 'doc2', 'doc3', 'doc4', 'doc5']
index = {}
for id, doc in enumerate(documents):
    filename = doc + ".txt"
    with open(filename, 'r') as fp:
        data = "".join(fp.readlines())
        data = data.lower()
        ext_words = re.findall(r"([a-z0-9-]+)", data)
        for pos, word in enumerate(ext_words):
            if word[-1] == 's':
                if word[:-1] in index:
                    word = word[:-1]
                elif word[:-2] in index:
                    word = word[:-2]

            if word not in index:
                index[word] = {"freq": 1, "listing": [(id+1, pos)]}
            else:
                index[word]["freq"] += 1
                index[word]["listing"].append((id+1, pos))
from collections import OrderedDict
index = OrderedDict(sorted(index.items()))
with open("inverted.txt", 'w') as fp:
    for key in index:
        print(f"{key} : {index[key]}")
        fp.write(f"{key} : {index[key]}\n")

a : {'freq': 2, 'listing': [(1, 2), (3, 2)]}
and : {'freq': 1, 'listing': [(5, 2)]}
applications : {'freq': 2, 'listing': [(1, 8), (4, 7)]}
be : {'freq': 1, 'listing': [(4, 3)]}
beautiful : {'freq': 1, 'listing': [(2, 0)]}
can : {'freq': 1, 'listing': [(4, 2)]}
crawling : {'freq': 1, 'listing': [(5, 3)]}
for : {'freq': 4, 'listing': [(1, 5), (2, 4), (3, 5), (4, 5)]}
framework : {'freq': 1, 'listing': [(1, 4)]}
is : {'freq': 4, 'listing': [(1, 1), (2, 2), (3, 1), (5, 5)]}
it : {'freq': 1, 'listing': [(3, 0)]}
java : {'freq': 1, 'listing': [(4, 0)]}
package : {'freq': 1, 'listing': [(3, 4)]}
pages : {'freq': 1, 'listing': [(3, 8)]}
parsing : {'freq': 1, 'listing': [(3, 6)]}
portable : {'freq': 1, 'listing': [(1, 3)]}
programming : {'freq': 1, 'listing': [(4, 1)]}
python : {'freq': 1, 'listing': [(3, 3)]}
scraping : {'freq': 2, 'listing': [(2, 6), (5, 0)]}
selenium : {'freq': 1, 'listing': [(1, 0)]}
soup : {'freq': 1, 'listing': [(2, 1)]}
testing : {'freq': 1, 'listing': [(1, 6)]}
the : {'freq': 1, 'listing': [(3, 7)]}
used : {'freq': 1, 'listing': [(4, 4)]}
useful : {'freq': 2, 'listing': [(2, 3), (5, 6)]}
web : {'freq': 5, 'listing': [(1, 7), (2, 5), (4, 6), (5, 1), (5, 4)]}
```

QUESTION 2

Part - 1

```
import re
documents = ['doc1', 'doc2', 'doc3', 'doc4', 'doc5']
index = {}
for id, doc in enumerate(documents):
    filename = doc + ".txt"
    with open(filename, 'r') as fp:
        data = "".join(fp.readlines())
        data = data.lower()
        #print(len(data))
    if re.findall(r"\bselenium\b", data) and re.findall(r"\bweb\b", data):
        print("Match found in", filename)
        ext_words1 = re.findall(r"\bselenium\b", data)
        ext_words2 = re.findall(r"\bweb\b", data)
        for pos, word in enumerate(ext_words1):
            if word not in index:
                index[word] = {"freq": 1, "listing": [(id+1, pos)]}
            else:
```



```

        index[word]['freq']+=1
        index[word]['listing'].append((id+1,pos))
    for pos,word in enumerate(ext_words2):
        if word not in index:
            index[word]={ "freq":1, "listing": [(id+1,pos)] }
        else:
            index[word]['freq']+=1
            index[word]['listing'].append((id+1,pos))
    else:
        print("No match found in", filename)
print("\n")
from collections import OrderedDict
index = OrderedDict(sorted(index.items()))
with open("inverted.txt",'w') as fp:
    print("Answer: \n")
    for key in index:
        print(f"{key} : {index[key]}")
        fp.write(f"{key} : {index[key]}\n")

```

```

Match found in doc1.txt
No match found in doc2.txt
No match found in doc3.txt
No match found in doc4.txt
No match found in doc5.txt

```

Answer:

```

selenium : {'freq': 1, 'listing': [(1, 0)]}
web : {'freq': 1, 'listing': [(1, 0)]}

```

QUESTION 2

Part - 2

```

[ ] import re
documents =['doc1','doc2','doc3','doc4','doc5']
index={}
for id,doc in enumerate(documents):
    filename = doc+".txt"
    with open(filename,'r') as fp:
        data = "".join(fp.readlines())
        data = data.lower()
        #print(len(data))
        ext_words = re.findall(r"\bsoup\b", data)
        for pos,word in enumerate(ext_words):
            if word not in index:
                index[word]={ "freq":1, "listing": [(id+1,pos)] }
            else:
                index[word]['freq']+=1
                index[word]['listing'].append((id+1,pos))
from collections import OrderedDict
index = OrderedDict(sorted(index.items()))
with open("inverted.txt",'w') as fp:
    for key in index:
        print(f"{key} : {index[key]}")
        fp.write(f"{key} : {index[key]}\n")

```

```
soup : {'freq': 1, 'listing': [(2, 0)]}
```

QUESTION 2

Part - 3

```

[ ] import re
documents =['doc1','doc2','doc3','doc4','doc5']
index={}
for id,doc in enumerate(documents):
    filename = doc+".txt"
    with open(filename,'r') as fp:
        data = "".join(fp.readlines())
        data = data.lower()
        #print(len(data))
        ext_words1 = re.compile(r"\bpython\b | \bjava\b",flags=re.I | re.X)
        ext_words2=ext_words1.findall(data)
        for pos,word in enumerate(ext_words2):
            if word not in index:
                index[word]={ "freq":1, "listing": [(id+1,pos)] }
            else:
                index[word]['freq']+=1
                index[word]['listing'].append((id+1,pos))
from collections import OrderedDict

```

```

from collections import OrderedDict
index = OrderedDict(sorted(index.items()))
with open("inverted.txt", 'w') as fp:
    for key in index:
        print(f"{key} : {index[key]}")
        fp.write(f"{key} : {index[key]}\n")

```

```

java : {'freq': 1, 'listing': [(4, 0)]}
python : {'freq': 1, 'listing': [(3, 0)]}

```

QUESTION 2

Part - 4

```

import re
documents = ['doc1', 'doc2', 'doc3', 'doc4', 'doc5']
index={}
for id, doc in enumerate(documents):
    filename = doc+".txt"
    with open(filename, 'r') as fp:
        data = "".join(fp.readlines())
        data = data.lower()
        #print(len(data))
        if re.findall(r"\bweb\b", data) and re.findall(r"\bcraw\b", data):
            print("Match found in", filename)
            ext_words1 = re.findall(r"\bweb\b", data)
            ext_words2 = re.findall(r"\bcraw\b", data)
            for pos, word in enumerate(ext_words1):
                if word not in index:
                    index[word]={'freq':1, "listing": [(id+1,pos)] }
                else:
                    index[word]['freq']+=1
                    index[word]['listing'].append((id+1,pos))
            for pos, word in enumerate(ext_words2):
                if word not in index:
                    index[word]={'freq':1, "listing": [(id+1,pos)] }
                else:
                    index[word]['freq']+=1
                    index[word]['listing'].append((id+1,pos))
            else:
                print("No match found in", filename)
                #break
    print("\n")
from collections import OrderedDict
index = OrderedDict(sorted(index.items()))
with open("inverted.txt", 'w') as fp:
    #print("Answer: \n")
    for key in index:
        print(f"{key} : {index[key]}")
        fp.write(f"{key} : {index[key]}\n")

```

```

No match found in doc1.txt
No match found in doc2.txt
No match found in doc3.txt
No match found in doc4.txt
No match found in doc5.txt

```