Lab Assignment - 6

In [3]:
```
d1= '''being an at from that are a they be or such for does same were she you the us am bee
d2= '''does with had was she or and has are from an could same be is they must you when the
d3= '''any where he who it so would do there they how it she could with a an have we more b
d4= '''he they more has how for may by same what I here from were a on there at we where do
d5= '''had many could will some would and I they the must been that such he who she then wh
d6= '''might be such he for he none how did I you we because should is will she at where it
```

In [5]:
```
# set(d1)
```

In [ ]:
```
# List of documents
documents = [d1, d2, d3, d4, d5, d6]

# Function to create the table
def create_table(documents):
    # Initialize an empty set for unique words
    unique_words = set()

    # Add unique words from each document to the set
    for doc in documents:
        unique_words.update(doc.split())

    # Creating the table
    table = {}
    for word in unique_words:
        table[word] = [1 if word in doc.split() else 0 for doc in documents]
    return table
```

In [7]:

```python
# Create the table
table = create_table(documents)

# Print the table
print("Unique Words\t", end="")
for i in range(1, 7):
    print(f"doc{i}\t", end="")
print()

for word, occurrences in table.items():
    print(f"{word}\t", end="")
    for occurrence in occurrences:
        print(f"{occurrence}\t", end="")
    print()
```

Unique Words	doc1	doc2	doc3	doc4	doc5	doc6
should 0	0	0	0	0	1	
an 1	1	1	0	0	0	
how 1	0	1	1	0	1	
you 1	1	1	0	0	1	
here 0	0	0	1	0	0	
because 0	0	0	0	0	1	
be 1	1	0	0	0	1	
same 1	1	0	1	0	0	
she 1	1	1	0	1	1	
been 1	0	0	0	1	1	
to 1	0	0	1	0	0	
had 0	1	1	0	1	0	
has 0	1	0	1	0	0	
are 1	1	0	0	1	0	
other 1	0	0	0	0	0	
at 1	0	0	1	0	1	
do 0	1	1	1	1	0	
on 0	0	0	1	0	1	
a 1	0	1	1	0	1	
us 1	0	0	0	0	1	
was 0	1	0	0	1	0	
none 1	0	0	1	0	1	
of 0	1	0	0	1	0	
with 0	1	1	1	0	0	
it 0	0	1	1	1	1	
would 0	0	1	0	1	0	
must 0	1	0	0	1	0	
he 0	0	1	1	1	1	
could 0	1	1	1	1	0	
then 1	0	0	0	1	0	
there 0	0	1	1	0	1	
some 0	1	0	1	1	0	
all 1	0	0	0	0	0	
they 1	1	1	1	1	0	
many 0	0	0	0	1	0	
may 1	0	0	1	0	0	
is 0	1	0	0	0	1	
the 1	1	0	0	1	1	
this 0	0	1	0	0	0	
by 0	0	1	1	0	0	
so 0	0	1	0	1	0	
will 0	0	0	0	1	1	
or 1	1	1	0	0	0	
were 1	1	0	1	0	0	
when 0	1	1	0	1	0	
more 0	0	1	1	0	0	
did 0	0	0	0	0	1	
in 0	0	1	0	1	1	
that 1	0	0	0	1	0	
am 1	1	0	0	0	0	
I 0	1	0	1	1	1	
we 1	0	1	1	0	1	
might 0	0	1	0	1	1	
and 0	1	0	0	1	0	
from 1	1	1	1	0	0	
being 1	1	0	0	1	0	
any 0	0	1	1	0	1	
have 0	0	1	0	0	1	
where 0	0	1	1	1	1	
does 1	1	1	1	0	1	
such 1	1	0	1	1	1	
what 0	1	0	1	0	0	
why 0	0	0	0	1	0	
who 0	0	1	0	1	0	
for 1	0	0	1	0	1	

In [37]:
```
import pandas as pd

# Document strings
d1 = '''being an at from that are a they be or such for does same were she you the us am be
d2 = '''does with had was she or and has are from an could same be is they must you when th
d3 = '''any where he who it so would do there they how it she could with a an have we more
d4 = '''he they more has how for may by same what I here from were a on there at we where d
d5 = '''had many could will some would and I they the must been that such he who she then w
d6 = '''might be such he for he none how did I you we because should is will she at where i

# List of document strings
d = [d1, d2, d3, d4, d5, d6]

# Initialize an empty set for unique words
unique_words = set()

# Add unique words from each document to the set
for i in d:
    unique_words.update(i.split())

# Create an empty DataFrame
df = pd.DataFrame(columns=['Unique Words'] + [f"d{i}" for i in range(1, 7)])

# Iterate over unique words to populate the DataFrame
for word in unique_words:
    occurrences = [1 if word in doc.split() else 0 for doc in documents]
    df.loc[len(df)] = [word] + occurrences

# Display the DataFrame
print(df)
```

```
   Unique Words  d1  d2  d3  d4  d5  d6
0        should   0   0   0   0   0   1
1            an   1   1   1   0   0   0
2           how   1   0   1   1   0   1
3           you   1   1   1   0   0   1
4          here   0   0   0   1   0   0
..          ...  ..  ..  ..  ..  ..  ..
60         such   1   1   0   1   1   1
61         what   0   1   0   1   0   0
62          why   0   0   0   0   1   0
63          who   0   0   1   0   1   0
64          for   1   0   0   1   0   1

[65 rows x 7 columns]
```

In [41]:
```
# df[(df['Unique Words'].str.contains('should')) & (df['Unique Words'].str.contains('for'))

df[(df['Unique Words'].str.contains('should')) & (df['Unique Words'].str.contains('you'))]
```

Out[41]:

Unique Words	d1	d2	d3	d4	d5	d6

In [44]:
```
df[(df['Unique Words'].str.contains('should'))],df[(df['Unique Words'].str.contains('you'))
```

Out[44]:
```
(  Unique Words  d1  d2  d3  d4  d5  d6
 0        should   0   0   0   0   0   1,
   Unique Words  d1  d2  d3  d4  d5  d6
 3           you   1   1   1   0   0   1)
```

In [45]:
```
# DataFrames for comparison
df_should = {'d1': 0, 'd2': 0, 'd3': 0, 'd4': 0, 'd5': 0, 'd6': 1}
df_you = {'d1': 1, 'd2': 1, 'd3': 1, 'd4': 0, 'd5': 0, 'd6': 1}

# Documents where 'should' and 'you' are present
should_docs = [doc for doc, presence in df_should.items() if presence == 1]
you_docs = [doc for doc, presence in df_you.items() if presence == 1]

# Columns where both 'should' and 'you' are present
common_columns = [doc for doc in should_docs if doc in you_docs]

# Display the common columns
print("Columns where both 'should' and 'you' are present:")
print(common_columns)
```

```
Columns where both 'should' and 'you' are present:
['d6']
```

In [49]:
```
query
```

Out[49]:
```
['should', 'you']
```

In [ ]:

In [ ]:

In [ ]:

In [24]:
```
def check_word_presence():
    word = input("Enter the word to check: ")
    presence_in_docs = []
    for i in range(1, 7):
        presence = "present" if word in df['Unique Words'].values and df[f"doc{i}"][df[df['
        presence_in_docs.append((i, presence))
    for doc_number, presence in presence_in_docs:
        print(f"The word '{word}' is {presence} in document {doc_number}.")

# Example usage
check_word_presence()
```

```
Enter the word to check: should
The word 'should' is not present in document 1.
The word 'should' is not present in document 2.
The word 'should' is not present in document 3.
The word 'should' is not present in document 4.
The word 'should' is not present in document 5.
The word 'should' is present in document 6.
```

In [25]:
```
def check_word_presence():
    word = input("Enter the word to check: ")
    for i in range(1, 7):
        if word in df['Unique Words'].values and df[f"doc{i}"][df[df['Unique Words'] == wor
            print(f"The word '{word}' is present in document {i}.")

# Example usage
check_word_presence()
```

```
Enter the word to check: do
The word 'do' is present in document 2.
The word 'do' is present in document 3.
The word 'do' is present in document 4.
The word 'do' is present in document 5.
```

In [ ]:
```
word = input("Enter the word to check: ")
for i in range(1, 7):
    if word in df['Unique Words'].values and df[f"doc{i}"][df[df['Unique Words'] == word].i
        print(f"The word '{word}' is present in document {i}.")
```

In [26]:
```
a = input()
for i in range(1, 7):
    if word in df['Unique Words'].values :
        print("yes")
```

```
Enter the word to check: do
yes
yes
yes
yes
yes
yes
```

In [32]:
```
word = input("Enter the word to check: ")
for i in range(1, 7):
    index = df[df['Unique Words'] == word].index
    if word in df['Unique Words'].values and df.iloc[index[0], i] == 1:
        print(i)
```

```
Enter the word to check: do
2
3
4
5
```

In [35]:
```
import pandas as pd

# Document strings
d1 = '''being an at from that are a they be or such for does same were she you the us am be
d2 = '''does with had was she or and has are from an could same be is they must you when th
d3 = '''any where he who it so would do there they how it she could with a an have we more
d4 = '''he they more has how for may by same what I here from were a on there at we where d
d5 = '''had many could will some would and I they the must been that such he who she then w
d6 = '''might be such he for he none how did I you we because should is will she at where i

# List of document strings
documents = [d1, d2, d3, d4, d5, d6]

# Create a dictionary to store the document names where each word is present
word_presence = {}

# Iterate over each document
for doc_num, doc_content in enumerate(documents, start=1):
    # Split the document content into words
    words = doc_content.split()
    # Iterate over each word in the document
    for word in words:
        # Check if the word is already present in the dictionary
        if word in word_presence:
            # If present, append the current document name to the list
            word_presence[word].append(f"d{doc_num}")
        else:
            # If not present, create a new list with the current document name
            word_presence[word] = [f"d{doc_num}"]

# Create a pandas DataFrame from the word_presence dictionary
df = pd.DataFrame(word_presence.items(), columns=['Word', 'Documents'])

# Display the DataFrame
print(df)
```

```
       Word         Documents
0     being      [d1, d2, d5]
1        an      [d1, d2, d3]
2        at      [d1, d4, d6]
3      from  [d1, d2, d3, d4]
4      that          [d1, d5]
..      ...               ...
60     will          [d5, d6]
61      why              [d5]
62      did              [d6]
63  because              [d6]
64   should              [d6]

[65 rows x 2 columns]
```

In [ ]: