In [ ]:

```
'''
import xml.etree.ElementTree
e = xml.etree.ElementTree.parse(url)


from bs4 import BeautifulSoup

y=BeautifulSoup(e)

import requests
import xml.etree.ElementTree as ET

r = requests.get(url)
root = ET.fromstring(r.text)

#from bs4 import BeautifulSoup

#y=BeautifulSoup(r)

print (r)



import json

from lxml import etree


import requests
import xml.etree.ElementTree as ET

r = requests.get(url)
root = ET.fromstring(r.text)


dom = etree.parse(r)
# load XSLT
transform = etree.XSLT(etree.fromstring(XSL))

# apply XSLT on loaded dom
json_text = str(transform(dom))

# json_text contains the data converted to JSON format.
# you can use it with the JSON API. Example:
data = json.loads(json_text)
print(data)

'''

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd
```

```
metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea


df1 = pd.read_csv(metadata_results)
#print(df1['filename'])

sample_jackets = ['CHRG-105hhrg40050']
count = 0
for jacket in df1['filename']:

   # try:
        print (count)
        url = 'https://api.govinfo.gov/packages/'+jacket+'/mods?&api_key=XNEgGxjbEszIMyIeni

        r = requests.get(url)

        with open('data.xml', 'w') as f:
            f.write(r.text)

        with open("data.xml", 'r') as f:
            xmlString = f.read()

        #print ("XML input (data.xml):")
        #print(xmlString)

        jsonString = json.dumps(xmltodict.parse(xmlString), indent=4)

        jsonObj = json.loads(jsonString)

        #print("\nJSON output(output.json):")
        #print(jsonString)

        #with open("output.json", 'w') as f:
        #    f.write(jsonString)


        witnesses = []
        witness_count = 0
        try:
            if "witness" in jsonObj["mods"]["extension"][2]:
                for witness in (jsonObj["mods"]["extension"][2]["witness"]):
                    witnesses.append(witness+'\n')
                    witness_count += 1
        except:
            witnesses.append ("Not found\n")

        count = count + 1


        print ("".join(witnesses))



        with open(metadata_results,'r') as csvinput:
            with open(metadata_results_new, 'a') as csvoutput:
                writer = csv.writer(csvoutput, lineterminator='\n')
                reader = csv.reader(csvinput)

                all = []
                row = next(reader)
                row.append('Witnesses & Affiliattions')
                all.append(row)

                for row in reader:
                    row.append("".join(witnesses))
                    all.append(row)

                writer.writerows(all)

        if (count > 2):
            break


    #except:
        #count = count + 1
     #   continue
```

Congressional committee name:

In [ ]:

```
print (jsonObj["mods"]["name"][0]["namePart"])
```

Witnesses:

In [ ]:

```
witness_count = 0
if "witness" in jsonObj["mods"]["extension"][2]:
    for witness in (jsonObj["mods"]["extension"][2]["witness"]):
        print (witness)
        witness_count += 1
else:
    print ("No witness information found")
```

Affiliations:

In [ ]:

```
nameAff = {}
for name in (jsonObj["mods"]["name"]):
    if name["@type"] == "personal" and "affiliation" in name:
        nameAff[name['namePart']] = name['affiliation']

for i in nameAff.items():
    print (i[0] + '\t' + i[1])
```

In [ ]:

```
# Metadata_results
# Committee number column  - from individual csv

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csv = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro

committees = {}

df1 = pd.read_csv(metadata_results)
#print(df1['filename'])


sample_jackets = [ 'CHRG-115hhrg27211']
count = 0
for jacket in df1['filename']:

    try:
        #print (count)

        #if (count > 50):
        #     break

        count = count + 1

        df2 = pd.read_csv(results_csv+jacket+'.csv')

        committees[jacket] = df2['committees'].iloc[0]


    except:
        count = count + 1
        continue

print (committees)
```

In [ ]:

```
# Metadata_results
# Committee number column  - from individual csv

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csv = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro


df1 = pd.read_csv(metadata_results)
#print(df1['filename'])

sample_jackets = [ 'CHRG-115hhrg27211']
count = 0

with open(metadata_results,'r') as csvinput:
            with open(metadata_results_new, 'w') as csvoutput:
                writer = csv.writer(csvoutput, lineterminator='\n')
                reader = csv.reader(csvinput)

                all = []
                row = next(reader)
                row.append('Committees')
                all.append(row)

                for row in reader:


                    try:

                        if ( not math.isnan(committees[row[5]]) ):
                                row.append(committees[row[5]])

                        else:
                            row.append("-")
                    except:
                                row.append("-")

                    all.append(row)

                writer.writerows(all)
```

In [ ]:

```
# Individual CSVs
# Affiliations

import sys
import csv

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings


df1 = pd.read_csv(metadata_results)
#print(df1['filename'])

sample_jackets = ['CHRG-115hhrg27211']
count = 0

files = set(os.listdir(results_csvs)) - set(os.listdir(results_csvs_new))

for file in files:
    try:

        url = 'https://api.govinfo.gov/packages/'+file.strip()[:-4]+'/mods?&api_key=qv508dp

        #print (url)
        r = requests.get(url)

        with open('data.xml', 'w', encoding="utf8") as f:
            f.write(r.text)

        with open("data.xml", 'r', encoding="utf8") as f:
            xmlString = f.read()

        #print ("XML input (data.xml):")
        #print(xmlString)

        jsonString = json.dumps(xmltodict.parse(xmlString), indent=4)
        jsonObj = json.loads(jsonString)

        with open(results_csvs+file,'r', encoding="utf8") as csvinput:
                with open(results_csvs_new+file, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    row.append('Full name')
                    row.append('Affiliation')
                    all.append(row)
                    #print (row)
                    #try:

                    for row in reader:


                        try:

                            if ( row[-1] == "Yes"):
                                    row.append("".join(row[5].split(",")[:2]).strip())
                                    row.append("".join(row[5].split(",")[2:]).strip())
                            else:

                                try:
                                    nameAff = {}
                                    for name in (jsonObj["mods"]["name"]):
                                        if name["@type"] == "personal" and "affiliation" in
                                            nameAff[name['namePart']] = name['affiliation']

                                    added = False
                                    for i in nameAff.items():
                                        if (fuzz.token_sort_ratio(i[0], row[5].strip()) > 8
                                                row.append(i[0])
                                                row.append(i[1])
                                                added = True
                                                break

                                    if(not added):
                                        row.append(row[5].strip())
                                        row.append("-")
                                except:
                                    row.append(row[5].strip())
                                    row.append("-")

                        except:
                                    row.append(row[5].strip())
                                    row.append("-")

                        all.append(row)

                    #except:
                    #    writer.writerows(all)
                    #    continue
                    writer.writerows(all)
    except:
        continue
```

In [ ]:

```
# Metadata_results
# Witness names & Affiliations, Members of the congress

import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing


df1 = pd.read_csv(metadata_results)
#print(df1['filename'])

sample_jackets = ['CHRG-115hhrg27211']
count = 0

#files = set(os.listdir(results_csvs)) - set(os.listdir(results_csvs_new))

with open(metadata_results,'r', encoding="utf8") as csvinput:
        with open(metadata_results_new, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)
                    row.append('Witnesses')
                    row.append('Members of the congress')
                    row.append('File exists')
                    all.append(row)
                    #print (row)
                    #try:

                    for row in reader:

                            #try:
                            if (row[6].strip()+'.csv' in os.listdir(results_csvs)):
                                print (row[6].strip()+'.csv')
                                file = pd.read_csv(results_csvs + row[6].strip() +'.csv')

                                #print (file.head())
                                witnesses = []
                                members = []


                                for index, row1 in file.iterrows():
                                    #print (row1['Witness'])
                                    temp = ''
                                    if (row1['Witness'].strip() == "Yes"):
                                        if (str(row1['Full name']).strip() != 'NA' and str(
                                            temp = str(row1['Full name'])
                                            if (str(row1['Affiliation']).strip() != 'NA' ar
                                                temp += ' : ' + str(row1['Affiliation']).st
                                                witnesses.append(temp)
                                            else:
                                                witnesses.append(temp + ';\n')
                                    else:
                                        if (str(row1['Full name']).strip() != 'NA' and str(
                                            temp = str(row1['Full name'])
                                            if (str(row1['Affiliation']).strip() != 'NA' ar
                                                temp += ' : ' + str(row1['Affiliation']).st
                                                members.append(temp)
                                            else:
                                                members.append(temp + ';\n')

                                #print (witnesses)

                                witnesses = [x for x in witnesses if str(x) != 'nan;']
                                members = [x for x in members if str(x) != 'nan;']

                                witnesses = set(witnesses)
                                members = set(members)


                                if (len(witnesses) == 0):
                                    row.append('-')
                                else:
                                    row.append("".join(witnesses).strip())

                                if (len(members) == 0):
                                    row.append('-')
                                else:
                                    row.append("".join(members).strip())

                                row.append("Yes")
                                all.append(row)

                            else:
                                    row.append('-')
                                    row.append('-')
                                    row.append("No")
                                    all.append(row)

                            #except:
                            #    row.append("-")
                            #    row.append("-")

                            #    all.append(row)
                            #    continue
                    #except:
                    #    writer.writerows(all)
                    #    continue
                    writer.writerows(all)
```

In [ ]:

```
# GPO agencies
# Individual CSVs

import csv
import pandas as pd
gpo = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_server


metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings


df = pd.read_csv(gpo)
agencies = []
for i in (df['Agency']):
    temp = i.replace('U.S.', 'United States')
    temp = temp.replace('U.S', 'United States')
    temp = temp.replace('Dep.', 'Department')

    agencies.append(temp)


#print (set(agencies))

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

file = pd.read_csv(sample_csvs + 'CHRG-104hhrg37344' +'.csv')
for index, row1 in file.iterrows():
    if (row1['Witness'] == "Yes"):
        max_score = 0
        for i in (set(agencies)):
            score = fuzz.token_set_ratio( i.lower(), row1['Affiliation'].lower())
            if (score > max_score):
                max_score = score
                agency = i
        print ( row1['Affiliation'] + ' : ' + agency + '\t' + str(max_score))
```

In [ ]:

```
# metadata_results_new
# Remove "nan"


import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csv = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro


df1 = pd.read_csv(metadata_results)
#print(df1['filename'])

sample_jackets = [ 'CHRG-115hhrg27211']
count = 0

with open(metadata_results,'r',encoding="utf8") as csvinput:
            with open(metadata_results_new, 'w',encoding="utf8") as csvoutput:
                writer = csv.writer(csvoutput, lineterminator='\n')
                reader = csv.reader(csvinput)

                all = []
                row = next(reader)

                all.append(row)

                for row in reader:

                    row[-2] = "\n".join(  list(filter(None, row[-2].replace('nan;','').spli

                    if(row[-2].strip() == ''):
                        row[-2] = '-'

                    all.append(row)

                writer.writerows(all)
```

In [ ]:

```
# Downloading API urls in json format to the local DB

import requests
import os
import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

APIs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

#files = set(os.listdir(results_csvs)) - set(os.listdir(results_csvs_new))

df1 = pd.read_csv(metadata_results)
#print(df1['filename'])


sample_jackets = ['CHRG-105hhrg40050']
count = 0
for jacket in df1['Filename']:

    try:
        #print (set(os.listdir(APIs)))
        #print (jacket+".json")
        if jacket+".json" not in set(os.listdir(APIs)):
            url = 'https://api.govinfo.gov/packages/'+jacket+'/mods?&api_key=XNEgGxjbEszIMy

            r = requests.get(url)

            with open('data.xml', 'w' , encoding="utf8") as f:
                f.write(r.text)

            with open("data.xml", 'r' , encoding="utf8") as f:
                xmlString = f.read()

            #print ("XML input (data.xml):")
            #print(xmlString)

            jsonString = json.dumps(xmltodict.parse(xmlString), indent=4)

            jsonObj = json.loads(jsonString)

            #print("\nJSON output(output.json):")
            #print(jsonString)

            file = APIs + jacket+ ".json"

            with open(file, 'w', encoding="utf8") as f:
                f.write(jsonString)

    except:
        print(jacket)
```

In [ ]:

```
# Downloading full text in .txt format to the local DB

import os
import urllib.request
import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

FullText = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_s
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

#files = set(os.listdir(results_csvs)) - set(os.listdir(results_csvs_new))

df1 = pd.read_csv(metadata_results)
#print(df1['filename'])


sample_jackets = ['CHRG-105hhrg40050']
count = 0
for jacket in df1['Filename']:

    try:
        #print (set(os.listdir(APIs)))
        #print (jacket+".json")
        if jacket+".txt" not in set(os.listdir(FullText)):

            url = 'https://api.govinfo.gov/packages/'+jacket+'/granules/'+jacket+'/htm?api_

            file = FullText + jacket + ".txt"

            urllib.request.urlretrieve(url, file)


    except:
        print(jacket)
```

In [ ]:

```
# Read the file in local DB

file = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

file_lines = open(file).readlines()
print (file_lines[:20])
```

In [ ]:

```
# Individual CSVs
# heldDate extraction

import json
from pprint import pprint

import sys
import csv

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

APIs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve


df1 = pd.read_csv(metadata_results)
#print(df1['filename'])

sample_jackets = ['CHRG-115hhrg27211']
```

```
count = 0

files = set(os.listdir(results_csvs)) - set(os.listdir(results_csvs_new))

for file in os.listdir(results_csvs):

    try:

        #with open(APIs+file, 'r') as f:
        #    xmlString = f.read()

        #print ("XML input (data.xml):")
        #print(xmlString)

        file = file.replace('.csv','.json')

        with open(APIs+file) as data_file:
            jsonObj = json.load(data_file)
        #print(jsonObj)

        file = file.replace('.json','.csv')

     #   if file == 'CHRG-100shrg83712.csv' or file == 'CHRG-102hhrg67539.csv' or file == '
      #       continue
        with open(results_csvs+file,'r', encoding="utf8") as csvinput:
                with open(results_csvs_new+file, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    row.append('heldDate')
                    all.append(row)
                    #print (row)
                    #try:

                    for row in reader:
                        try:
                            heldDate = []
                            added = False
                            exists = False

                            for item in (jsonObj["mods"]["extension"]):
                                        #for item in extension:
                                            #print (item)
                                            if "heldDate" in item:
                                                exists = True
                                                if isinstance(item["heldDate"], list):
                                                    for date in item["heldDate"]:
                                                        heldDate.append(date)
                                                        added = True
                                                        #print (heldDate)
                                                else:
                                                    row.append(item["heldDate"])
                                                    #print (item["heldDate"])
                                                    break

                            if exists == False:
                                row.append("-")
                            if added:
                                row.append(";\n".join(heldDate))
                                #break

                        except:
                            row.append("-")

                        all.append(row)

                    #except:
                    #    writer.writerows(all)
                    #    continue
                    writer.writerows(all)
    except:
        print (file)
```

In [ ]:

```
# metadata_results
# heldDate extraction

import json
from pprint import pprint

import sys
import csv

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

APIs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve


df1 = pd.read_csv(metadata_results)
#print(df1['filename'])

sample_jackets = ['CHRG-115hhrg27211']
```

```
count = 0

files = set(os.listdir(results_csvs)) - set(os.listdir(results_csvs_new))


with open(metadata_results,'r', encoding="utf8") as csvinput:
                with open(metadata_results_new, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    try:


                        all = []
                        row = next(reader)

                        row.append('heldDate')
                        all.append(row)
                        #print (row)
                        #try:


                        for row in reader:
                            try:


                                #if (row[6].strip()+'.csv' in os.listdir(results_csvs)):
                                    #print (row[6].strip()+'.csv')
                                file = row[6].strip()

                                file = file + '.json'

                                with open(APIs+file) as data_file:
                                        jsonObj = json.load(data_file)
                                    #print(jsonObj)

                                heldDate = []
                                added = False
                                exists = False

                                for item in (jsonObj["mods"]["extension"]):
                                            #for item in extension:
                                                #print (item)
                                                if "heldDate" in item:
                                                    exists = True
                                                    if isinstance(item["heldDate"], list):
                                                        for date in item["heldDate"]:
                                                            heldDate.append(date)
                                                            added = True
                                                            #print (heldDate)
                                                    else:
                                                        row.append(item["heldDate"])
                                                       # print (item["heldDate"])
                                                        break

                                if exists == False:
                                    row.append("-")
                                if added:
                                    row.append(";\n".join(heldDate))
                                    #break
                            except:
                                row.append("-")

                            all.append(row)

                        #except:
                        #    writer.writerows(all)
                        #    continue
                        writer.writerows(all)

                    except:
                        print (file)
```

In [ ]:

```
# GPO agencies for sample 500 CSVs
# Individual CSVs

import csv
import pandas as pd
gpo = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_server
gpo2 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

sample500 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_

sample500GPOOutput = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_heari

df1 = pd.read_csv(sample500)
#print(df1['filename'])

df = pd.read_csv(gpo2)
agencies = []
for i in (df['Agency']):
    temp = i.replace('U.S.', 'United States')
    temp = temp.replace('U.S', 'United States')
    temp = temp.replace('Dep.', 'Department')

    agencies.append(temp)


#print (set(agencies))

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#file = pd.read_csv(sample_csvs + 'CHRG-105hhrg40051' +'.csv')

for file in df1['filename']:

    try:

                #print ( row1['Affiliation'] + ' : ' + agency + '\t' + str(max_score))

        with open(results_csvs+file+'.csv','r', encoding="utf8") as csvinput:
                    with open(sample500GPOOutput+file+'.csv', 'w+', encoding="utf8") as csv
                        writer = csv.writer(csvoutput, lineterminator='\n')
                        reader = csv.reader(csvinput)

                        all = []
                        row = next(reader)

                        row.append('Government agencies')
                        all.append(row)
                        #print (row)
                        #try:

                        for row in reader:

                            file1 = pd.read_csv(results_csvs + file +'.csv')

                            max_score = 0
                            agency = '-'
                            #print (row[18])
                            if (str(row[16]).strip() == "Yes"):
                                    max_score = 0
                                    agency = '-'
                                    for i in (set(agencies)):
                                        score = fuzz.token_set_ratio( i.lower(), row[18].lo
                                        if (score > max_score):
                                            max_score = score
                                            agency = i


                            if max_score == 100:
                                row.append(agency)
                            else:
                                row.append(agency)

                            all.append(row)

                        #except:
                        #    writer.writerows(all)
                        #    continue
                        writer.writerows(all)
    except:
        print (file)
```

In [ ]:

```
# Sentiment analysis for sample 500 CSVs
# Individual CSVs

import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')

import csv
import pandas as pd
gpo = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_server
gpo2 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

sample500 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_

sample500GPOOutput = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_heari

sample500SAOutput = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearin

df1 = pd.read_csv(sample500)
#print(df1['filename'])

sid = SentimentIntensityAnalyzer()

#print (set(agencies))

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#file = pd.read_csv(sample_csvs + 'CHRG-105hhrg40051' +'.csv')

for file in df1['filename']:

    try:

                #print ( row1['Affiliation'] + ' : ' + agency + '\t' + str(max_score))

        with open(results_csvs+file+'.csv','r', encoding="utf8") as csvinput:
                    with open(sample500SAOutput+file+'.csv', 'w+', encoding="utf8") as csvo
                        writer = csv.writer(csvoutput, lineterminator='\n')
                        reader = csv.reader(csvinput)

                        all = []
                        row = next(reader)

                        row.append('Sentiment analysis')
                        all.append(row)
                        #print (row)
                        #try:
```

```
                    #print (ss)
                    #print (max(ss, key=ss.get))
                    #break
                    for row in reader:

                        #df2 = pd.read_csv(results_csvs+file+'.csv')

                        #print (df2['cleaned'])
                        ss = sid.polarity_scores(row[12])

                        del (ss['compound'])

                        #print (row[12])

                        if ( max(ss, key=ss.get) == 'neu'):
                            row.append('Neutral')


                        if ( max(ss, key=ss.get) == 'neg'):
                            row.append('Negative')


                        if ( max(ss, key=ss.get) == 'pos'):
                            row.append('Positive')

                        all.append(row)

                    #except:
                    #    writer.writerows(all)
                    #    continue
                    writer.writerows(all)
except:
    print (file)
```

In [ ]:

```
# Metadata_results
# Witness names & Affiliations, Members of the congress from FULL Texts  - Scrapped Witness

import re
import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

FullTexts = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_


df1 = pd.read_csv(metadata_results)
#print(df1['filename'])

sample_jackets = ['CHRG-115hhrg27211']
count = 0

#files = set(os.listdir(results_csvs)) - set(os.listdir(results_csvs_new))

countWitness = 0
with open(metadata_results,'r', encoding="utf8") as csvinput:
        with open(metadata_results_new, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    row.append('Scrapped witnesses')
                    all.append(row)

                    for row in reader:

                            #try:
                            if (row[9]!='Appropriation' and row[9]!='Nomination') and row[1

                                    #print(row)
                                    if row[6]+'.txt' in set(os.listdir(FullTexts)):
                                        filename = FullTexts+row[6]+'.txt'
                                        lines = open(filename, "r", encoding="utf8").readli

                                        #print (lines)
                                        strippedLines = []
                                        for line in lines:
                                            #print (line.strip())
                                            strippedLines.append(line.strip())


                                        if ('C O N T E N T S' in strippedLines and 'Stateme


                                                startingIndex = strippedLines.index('Statem
                                                #print (startingIndex)
                                                witness = []

                                                #print ('\n'+row[6])
                                                #print (lines)
                                                witnessStr = []
                                                firstHit = 0
                                                for i in range(startingIndex+1, len(lines))
                                                    if '    ' in lines[i]:

                                                        if lines[i].strip() == '':
                                                            continue

                                                        if re.search(r"\.(\.)+( *)[0-9]*(\*
                                                            if(firstHit == 0):
                                                                x = re.sub('\.(\.)+( *)[0-9
                                                                witness.append(x.strip()+'\
                                                                firstHit = 1

                                                        elif ';' in lines[i]:
                                                            witness.append(lines[i].strip()
                                                            witness.append(lines[i].strip()
                                                            firstHit = 0

                                                        else:
                                                            witness.append(lines[i].strip()
                                                            firstHit = 0
                                                    else:
                                                        break


                                                #print ("".join(witness))
                    if(len("".join(witness)) < 6000):
                        row.append("".join(witness))
                        row[14] = 'Refer column S'
                        countWitness += 1


elif ('C O N T E N T S' in strippedLines and 'STATE
        #countWitness += 1

        startingIndex = strippedLines.index('STATEM
        #print (startingIndex)
        witness = []

        #print ('\n'+row[6])
        #print (lines)
        witnessStr = []
        firstHit = 0
        for i in range(startingIndex+1, len(lines))

                if 'APPENDIX' in lines[i] or 'Appen
                    break
                if 'Page' in lines[i]:
                    continue

                if lines[i].isupper():
                    break

                if lines[i].strip() == '':
                    continue

                if re.search(r"\.(\.)+( *)[0-9]*(\*
                    if(firstHit == 0):
                        x = re.sub('\.(\.)+( *)[0-9
                        witness.append(x.strip()+'\
                        firstHit = 1

                elif ';' in lines[i]:
                    witness.append(lines[i].split('
                    witness.append(lines[i].split('
                    firstHit = 0

                else:
                    witness.append(lines[i].strip()
                    firstHit = 0


        #print ("".join(witness))
        if(len("".join(witness)) < 6000):
            row.append("".join(witness))
            row[14] = 'Refer column S'
            countWitness += 1

elif ('CONTENTS' in strippedLines and 'TESTIMONY' i
        #countWitness += 1

        startingIndex = strippedLines.index('TESTIM
        #print (startingIndex)
        witness = []

        #print ('\n'+row[6])
        #print (lines)
```

```
    witnessStr = []
    firstHit = 0
    for i in range(startingIndex+1, len(lines))

        if 'APPENDIX' in lines[i] or 'Appen
          break
        if 'Page' in lines[i]:
          continue

        if lines[i].isupper():
          break

        if lines[i].strip() == '':
          continue

        if re.search(r"\.(\.)+( *)[0-9]*(\*
          if(firstHit == 0):
            x = re.sub('\.(\.)+( *)[0-9
            witness.append(x.strip()+'\
            firstHit = 1

        elif ';' in lines[i]:
          witness.append(lines[i].split('
          witness.append(lines[i].split('
          firstHit = 0

        else:
          witness.append(lines[i].strip()
          firstHit = 0


    #print ("".join(witness))
    if(len("".join(witness)) < 6000):
      row.append("".join(witness))
      row[14] = 'Refer column S'
      countWitness += 1

elif ('C O N T E N T S' in strippedLines and 'Testi
    #countWitness += 1

    startingIndex = strippedLines.index('Testin
    #print (startingIndex)
    witness = []

    #print ('\n'+row[6])
    #print (lines)
    witnessStr = []
    firstHit = 0
    for i in range(startingIndex+1, len(lines))

        if 'APPENDIX' in lines[i] or 'Appen
          break
        if 'Page' in lines[i]:
          continue

        if lines[i].isupper():
          break

        if lines[i].strip() == '':
          continue
```

```
                if re.search(r"\.(\.)+( *)[0-9]*(\*
                    if(firstHit == 0):
                        x = re.sub('\.(\.)+( *)[0-9
                        witness.append(x.strip()+'\
                        firstHit = 1

                elif ';' in lines[i]:
                    witness.append(lines[i].split('
                    witness.append(lines[i].split('
                    firstHit = 0

                else:
                    witness.append(lines[i].strip()
                    firstHit = 0


        #print ("".join(witness))
        if(len("".join(witness)) < 6000):
            row.append("".join(witness))
            row[14] = 'Refer column S'
            countWitness += 1

elif ('C O N T E N T S' in strippedLines and 'CHRON
        #countWitness += 1

        startingIndex = strippedLines.index('CHRONO
        #print (startingIndex)
        witness = []

        #print ('\n'+row[6])
        #print (lines)
        witnessStr = []
        firstHit = 0
        for i in range(startingIndex+1, len(lines))


                if lines[i].isupper():
                    break

                if lines[i].strip() == '':
                    continue

                if re.search(r"\.(\.)+( *)[0-9]*(\*
                    if(firstHit == 0):
                        x = re.sub('\.(\.)+( *)[0-9
                        witness.append(x.strip()+'\
                        firstHit = 1

                elif ';' in lines[i]:
                    witness.append(lines[i].split('
                    witness.append(lines[i].split('
                    firstHit = 0

                else:
                    witness.append(lines[i].strip()
                    firstHit = 0


        #print ("".join(witness))
        if(len("".join(witness)) < 6000):
            row.append("".join(witness))
```

```
                                                    row[14] = 'Refer column S'
                                                    countWitness += 1

                                        elif ('C O N T E N T S' in strippedLines and ('Pane
                                                #countWitness += 1

                                                if 'Panel I' in strippedLines:
                                                    startingIndex = strippedLines.index('Pa
                                                if 'PANEL I' in strippedLines:
                                                    startingIndex = strippedLines.index('PA

                                                #print (startingIndex)
                                                witness = []

                                                #print ('\n'+row[6])
                                                #print (lines)
                                                witnessStr = []
                                                firstHit = 0
                                                for i in range(startingIndex+1, len(lines))

                                                        if lines[i].strip == '----------':
                                                            break

                                                        if lines[i].strip() == '':
                                                            continue

                                                        if 'Panel' in lines[i]:
                                                            continue

                                                        if 'APPENDIX' in lines[i] or 'Apper
                                                            break
                                                        if 'Page' in lines[i]:
                                                            continue

                                                        if lines[i].isupper():
                                                            break

                                                        if re.search(r"\.(\.)+( *)[0-9]*(\*
                                                            if(firstHit == 0):
                                                                x = re.sub('\.(\.)+( *)[0-9
                                                                witness.append(x.strip()+'\
                                                                firstHit = 1

                                                        elif ';' in lines[i]:
                                                            witness.append(lines[i].split('
                                                            witness.append(lines[i].split('
                                                            firstHit = 0

                                                        else:
                                                            witness.append(lines[i].strip()
                                                            firstHit = 0


                                                #print ("".join(witness))
                                                if(len("".join(witness)) < 6000):
                                                    row.append("".join(witness))
                                                    row[14] = 'Refer column S'
                                                    countWitness += 1

                                        elif ('C O N T E N T S' in strippedLines and ('Part
                                                #countWitness += 1
                                                if 'Participants' in strippedLines:
                                                    startingIndex = strippedLines.index('Pa

                                                #print (startingIndex)
                                                witness = []

                                                #print ('\n'+row[6])
                                                #print (lines)
                                                witnessStr = []
                                                firstHit = 0
                                                for i in range(startingIndex+2, len(lines))

                                                        if lines[i].strip == '----------':
                                                            break

                                                        if lines[i].strip() == '':
                                                            break

                                                        if 'Panel' in lines[i]:
                                                            continue

                                                        if 'APPENDIX' in lines[i] or 'Appen
                                                            break
                                                        if 'Page' in lines[i]:
                                                            continue

                                                        if lines[i].isupper():
                                                            break

                                                        if re.search(r"\.(\.)+( *)[0-9]*(\*
                                                            if(firstHit == 0):
                                                                x = re.sub('\.(\.)+( *)[0-9
                                                                witness.append(x.strip()+'\
                                                                firstHit = 1

                                                        elif ';' in lines[i]:
                                                            witness.append(lines[i].split('
                                                            witness.append(lines[i].split('
                                                            firstHit = 0

                                                        else:
                                                            witness.append(lines[i].strip()
                                                            firstHit = 0


                                                #print ("".join(witness))
                                                if(len("".join(witness)) < 6000):
                                                    row.append("".join(witness))
                                                    row[14] = 'Refer column S'
                                                    countWitness += 1


                                        elif ('C O N T E N T S' in strippedLines and 'State
                                                #countWitness += 1

                                                startingIndex = strippedLines.index('Statem
                                                #print (startingIndex)
                                                witness = []

                                                #print ('\n'+row[6])
                                                #print (lines)
                                                witnessStr = []
                                                firstHit = 0
                                                for i in range(startingIndex+1, len(lines))

                                                        if 'APPENDIX' in lines[i] or 'Apper
                                                            break
                                                        if 'Page' in lines[i]:
                                                            continue

                                                        if lines[i].isupper():
                                                            break

                                                        if lines[i].strip() == '':
                                                            continue

                                                        if re.search(r"\.(\.)+( *)[0-9]*(\*
                                                            if(firstHit == 0):
                                                                x = re.sub('\.(\.)+( *)[0-9
                                                                witness.append(x.strip()+'\
                                                                firstHit = 1

                                                        elif ';' in lines[i]:
                                                            witness.append(lines[i].split('
                                                            witness.append(lines[i].split('
                                                            firstHit = 0

                                                        else:
                                                            witness.append(lines[i].strip()
                                                            firstHit = 0


                                                #print ("".join(witness))
                                                if(len("".join(witness)) < 6000):
                                                    row.append("".join(witness))
                                                    row[14] = 'Refer column S'
                                                    countWitness += 1

                                        elif ('C O N T E N T S' in strippedLines and 'WITNE
                                                #countWitness += 1

                                                startingIndex = strippedLines.index('WITNES
                                                #print (startingIndex)
                                                witness = []

                                                #print ('\n'+row[6])
                                                #print (lines)
                                                witnessStr = []
                                                firstHit = 0
                                                for i in range(startingIndex+1, len(lines))

                                                        if 'APPENDIX' in lines[i] or 'Apper
                                                            break
                                                        if 'Page' in lines[i]:
                                                            continue

                                                        if lines[i].isupper():
                                                            break

                                                        if lines[i].strip() == '':
                                                            continue
```

```
                            if re.search(r"\.(\.)+( *)[0-9]*(\*
                              if(firstHit == 0):
                                x = re.sub('\.(\.)+( *)[0-9
                                witness.append(x.strip()+'\
                                firstHit = 1

                            elif ';' in lines[i]:
                              witness.append(lines[i].split('
                              witness.append(lines[i].split('
                              firstHit = 0

                            else:
                              witness.append(lines[i].strip()
                              firstHit = 0


                        #print ("".join(witness))
                        if(len("".join(witness)) < 6000):
                          row.append("".join(witness))
                          row[14] = 'Refer column S'
                          countWitness += 1

                    elif ('C O N T E N T S' in strippedLines and 'Witne
                        #countWitness += 1

                        startingIndex = strippedLines.index('Witnes
                        #print (startingIndex)
                        witness = []

                        #print ('\n'+row[6])
                        #print (lines)
                        witnessStr = []
                        firstHit = 0
                        for i in range(startingIndex+1, len(lines))

                            if 'APPENDIX' in lines[i] or 'Apper
                              break
                            if 'Page' in lines[i]:
                              continue

                            if lines[i].isupper():
                              break

                            if lines[i].strip() == '':
                              continue

                            if re.search(r"\.(\.)+( *)[0-9]*(\*
                              if(firstHit == 0):
                                x = re.sub('\.(\.)+( *)[0-9
                                witness.append(x.strip()+'\
                                firstHit = 1

                            elif ';' in lines[i]:
                              witness.append(lines[i].split('
                              witness.append(lines[i].split('
                              firstHit = 0

                            else:
                              witness.append(lines[i].strip()
                              firstHit = 0
```

```
        #print ("".join(witness))
        if(len("".join(witness)) < 6000):
            row.append("".join(witness))
            row[14] = 'Refer column S'
            countWitness += 1

elif ('THE FUTURE OF THE OSCE MEDITERRANEAN PARTNER
        #countWitness += 1

        startingIndex = strippedLines.index('WITNES
        #print (startingIndex)
        witness = []

        #print ('\n'+row[6])
        #print (lines)
        witnessStr = []
        firstHit = 0
        for i in range(startingIndex+1, len(lines))

                if 'APPENDIX' in lines[i] or 'Apper
                    break
                if 'Page' in lines[i]:
                    continue

                if lines[i].isupper():
                    break

                if lines[i].strip() == '':
                    continue

                if re.search(r"\.(\.)+( *)[0-9]*(\*
                    if(firstHit == 0):
                        x = re.sub('\.(\.)+( *)[0-9
                        witness.append(x.strip()+'\
                        firstHit = 1

                elif ';' in lines[i]:
                    witness.append(lines[i].split('
                    witness.append(lines[i].split('
                    firstHit = 0

                else:
                    witness.append(lines[i].strip()
                    firstHit = 0


        #print ("".join(witness))
        if(len("".join(witness)) < 6000):
            row.append("".join(witness))
            row[14] = 'Refer column S'
            countWitness += 1

elif ('C O N T E N T S' in strippedLines and 'Page'
        #countWitness += 1

        startingIndex = strippedLines.index('Page')
        #print (startingIndex)
        witness = []
```

```
                                                #print ('\n'+row[6])
                                                #print (lines)
                                                witnessStr = []
                                                firstHit = 0
                                                for i in range(startingIndex+1, len(lines))

                                                        if 'APPENDIX' in lines[i] or 'Apper
                                                            break
                                                        if 'Page' in lines[i]:
                                                            continue

                                                        if lines[i].isupper():
                                                            break

                                                        if lines[i].strip() == '':
                                                            continue

                                                        if re.search(r"\.(\.)+( *)[0-9]*(\*
                                                            if(firstHit == 0):
                                                                x = re.sub('\.(\.)+( *)[0-9
                                                                witness.append(x.strip()+'\
                                                                firstHit = 1

                                                        elif ';' in lines[i]:
                                                            witness.append(lines[i].split('
                                                            witness.append(lines[i].split('
                                                            firstHit = 0

                                                        else:
                                                            witness.append(lines[i].strip()
                                                            firstHit = 0


                                                #print ("".join(witness))

                                                if(len("".join(witness)) < 6000):
                                                    row.append("".join(witness))
                                                    row[14] = 'Refer column S'
                                                    countWitness += 1

                                        else:
                                            row.append('-')
                                            #row[14] = 'Refer column S'

                            else:
                                row.append('-')

                            #if countWitness !=0:
                            #    break
                            #except:
                            #    row.append("-")
                            #    row.append("-")
                            all.append(row)
                            #    all.append(row)
                            #    continue
                    #except:
                    #    writer.writerows(all)
                    #    continue
                    writer.writerows(all)

print(countWitness)
```

In [ ]:

```
# Metadata_results
# Witness names & Affiliations, Members of the congress from FULL Texts - Scrapped Witnesse

import re
import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

FullTexts = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_


df1 = pd.read_csv(metadata_results)
#print(df1['filename'])

sample_jackets = ['CHRG-115hhrg27211']
count = 0

#files = set(os.listdir(results_csvs)) - set(os.listdir(results_csvs_new))

countWitness = 0
```

```
gpo = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_server
gpo2 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

sample500 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_

sample500GPOOutput = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_heari

sample500SAOutput = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearin

#df1 = pd.read_csv(sample_csvs_new)
#print(df1['filename'])

#sid = SentimentIntensityAnalyzer()

#print (set(agencies))

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#file = pd.read_csv(sample_csvs + 'CHRG-105hhrg40051' +'.csv')

scrappedWD = {}


with open(metadata_results,'r', encoding="utf8") as csvinput:

                    reader = csv.reader(csvinput)

                    for row in reader:

                            #try:
                            if row[13] =='Refer column R' :

                                    scrappedWD[row[6]] = row[17]

for k, v in scrappedWD.items():
    print (v.split('\n'))
    break
```

In [ ]:

```
# Metadata_results
# Witness names & Affiliations, Members of the congress from FULL Texts - Scrapped Witnesse

import re
import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

FullTexts = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_


#df1 = pd.read_csv(metadata_results)
#print(df1['filename'])

sample_jackets = ['CHRG-115hhrg27211']
count = 0

#files = set(os.listdir(results_csvs)) - set(os.listdir(results_csvs_new))

countWitness = 0
```

```
gpo = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_server
gpo2 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_frc
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

sample500 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_

sample500GPOOutput = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_heari

sample500SAOutput = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearir

df1 = pd.read_csv(sample500)
#print(df1['filename'])

#sid = SentimentIntensityAnalyzer()

#print (set(agencies))

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#file = pd.read_csv(sample_csvs + 'CHRG-105hhrg40051' +'.csv')

for file in set(os.listdir(results_csvs)):
#for file in set(os.listdir(results_csvs)):
    #print (file)
    #print (set(os.listdir(results_csvs)))
    #file = file + '.csv'
    #if file in set(os.listdir(results_csvs)):
    file = file.replace('.csv','')

    with open(results_csvs+file+'.csv','r', encoding="utf8") as csvinput:
            with open(results_csvs_new+file+'.csv', 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    row.append('Scrapped witnesses')
                    all.append(row)


                    for row in reader:
                        hit = 0
                        if row[7].strip() in scrappedWD.keys():
                            tempWit = scrappedWD[row[7].strip()]
                            #print (tempWit)
                            name = row[3] +' '+row[5] +' '+ row[17]
                            for j in tempWit.split('\n'):
                                if fuzz.token_sort_ratio("".join(j.lower().split()[:4]), na
                                    row.append(j.strip())
                                    row[16] = 'Yes'
                                    hit = 1
                                    #break
                                    #print (fuzz.token_sort_ratio("".join(j.lower().split()
                                    #print ("".join(j.lower().split()[:4]))
                                    #print (name.lower())
                                    break

                            if hit == 0:
                                row.append('-')
                        else:
                            row.append('-')

                        all.append(row)

                    writer.writerows(all)
```

In [ ]:

```
# Cleaning witness, scrapped witness column

import re
import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

for file in set(os.listdir(results_csvs)):
    with open(results_csvs+file,'r', encoding="utf8") as csvinput:
        with open(results_csvs_new+file, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    all.append(row)

                    for row in reader:

                        if row[18].strip() == 'United States Senate' or row[18].strip() ==
                            row[16] = 'No'
                            row[20] = '-'

                        all.append(row)
                    writer.writerows(all)

print ('asdf')
```

In [ ]:

```
# Creating dictionary of acronyms and agencies

import os

import math
import csv
import pandas as pd

gpo2 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve


#print(df1['filename'])

df = pd.read_csv(gpo2)
agencies = []
acronyms = []

acroMap = {}

for i in (df['Agency']):
    agencies.append(i)

for i in (df['Alternate Name']):
    acronyms.append(i)

for i in acronyms:
    if not(pd.isnull(i)):
        index = acronyms.index(i)
        acroMap[i] = agencies[index]

print((acroMap.keys()))
```

In [ ]:

```
# Creating dictionary of acronyms and states

import os

import math
import csv
import pandas as pd

usstates = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_s


df = pd.read_csv(usstates,  header=None)
states= []
acronyms = []

acroMapStates = {}

for i in (df.iloc[:,1]):
    states.append(i)

for i in (df.iloc[:,2]):
    acronyms.append(i)

for i in acronyms:
    #if not(pd.isnull(i)):
        index = acronyms.index(i)
        acroMapStates[i] = states[index]

print((acroMapStates.keys()))
```

In [ ]:

```
# GPO agencies for individual CSVs
# Exact matching on agency names and acronyms, states, Inspector General

import re
import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math
import csv
import pandas as pd
gpo = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_server
gpo2 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
results_csvs_new1 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearin


sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

sample500 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_

sample500GPOOutput = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_heari

df1 = pd.read_csv(sample500)
#print(df1['filename'])

df = pd.read_csv(gpo2)
agencies = []
for i in (df['Agency']):
    temp = i.replace('U.S.', 'United States')
    temp = temp.replace('US', 'United States')
    temp = temp.replace('Dep.', 'Department')
    temp = temp.replace('Dept.', 'Department')
    temp = temp.replace('Dept', 'Department')
    temp = temp.replace('Assoc', 'Association')
    temp = temp.replace('Assoc.', 'Association')
    temp = temp.replace('Brd', 'Board')
    temp = temp.replace('Brd.', 'Board')
    temp = temp.replace('DC', 'District of Columbia')
    temp = temp.replace('D.C.', 'District of Columbia')

    temp = temp.replace('.,',' ')
    temp = temp.replace('.;',' ')
    temp = temp.replace('.-',' ')
    temp = temp.replace('.:',' ')
    temp = temp.replace('.,',' ')

    temp = temp.replace('.', '')

    for i in temp.split():
        if i in acroMap.keys():
            temp = temp.replace(i,acroMap[i])

    for i in temp.split():
        if i in acroMapStates.keys():
            temp = temp.replace(i,acroMapStates[i])

    agencies.append(temp)

JK = []
UA = []
Parent = []

for i in (df['JK Code']):
    JK.append(i)
for i in (df['UA Code']):
    UA.append(i)
for i in (df['Parent UA Code']):
    Parent.append(i)


#print (set(agencies))

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#file = pd.read_csv(sample_csvs + 'CHRG-105hhrg40051' +'.csv')

#for file in df1['filename']:

#    try:

                #print ( row1['Affiliation'] + ' : ' + agency + '\t' + str(max_score))
#agencies = agencies[:100]
for file in set(os.listdir(results_csvs)):
    if file not in set(os.listdir(results_csvs_new)):
        with open(results_csvs+file,'r', encoding="utf8") as csvinput:
            with open(results_csvs_new+file, 'w+', encoding="utf8") as csvoutput:
                        writer = csv.writer(csvoutput, lineterminator='\n')
                        reader = csv.reader(csvinput)

                        all = []
                        row = next(reader)

                        row.append('Agency')
                        row.append('JK code')
                        row.append('UA code')
                        row.append('Parent UA code')
                        row.append('US State')
                        row.append('Inspector General')
                        all.append(row)
                        #print (row)
                        #try:

                        for row in reader:

                            if row[16] == 'Yes':
                                max_score = 0
                                agency = '-'
                                jk = '-'
                                ua = '-'
                                parent = '-'
                                aff = row[18] +' '+row[20]

                                aff = aff.replace('U.S.', 'United States')
                                aff = aff.replace('US', 'United States')
                                aff = aff.replace('Dep.', 'Department')
                                aff = aff.replace('Dept.', 'Department')
                                aff = aff.replace('Dept', 'Department')
                                aff = aff.replace('Assoc', 'Association')
                                aff = aff.replace('Assoc.', 'Association')
                                aff = aff.replace('Brd', 'Board')
                                aff = aff.replace('Brd.', 'Board')
                                aff = aff.replace('DC', 'District of Columbia')
                                aff = aff.replace('D.C.', 'District of Columbia')

                                aff = aff.replace('.,',' ')
                                aff = aff.replace('.;',' ')
                                aff = aff.replace('.-',' ')
                                aff = aff.replace('.:',' ')
                                aff = aff.replace('.,',' ')

                                aff = aff.replace('.', '')

                                for i in aff.split():
                                    if i in acroMap.keys():
                                        aff = aff.replace(i,acroMap[i])

                                for i in aff.split():
                                    if i in acroMapStates.keys():
                                        aff = aff.replace(i,acroMapStates[i])

                                hit = 0
                                for i in ((agencies)):
                                    #score = fuzz.WRatio( i, aff )
                                    #if (score > max_score):
                                    if i in aff:
                                        #max_score = score
                                        agency = i
                                        index = agencies.index(i)
                                        jk = JK[index]
                                        ua = UA[index]
                                        parent = Parent[index]
                                        row.append(agency)
                                        row.append(jk)
                                        row.append(ua)
                                        row.append(parent)
                                        hit = 1
                                        break
                                '''
                                if max_score >= 90:
                                    row.append(agency)
                                    row.append(jk)
                                    row.append(ua)
                                    row.append(parent)

                                else:
                                    row.append('-')
                                    row.append('-')
                                    row.append('-')
                                    row.append('-')
                                '''

                                if hit == 0:
                                    row.append('-')
                                    row.append('-')
                                    row.append('-')
                                    row.append('-')

                                states = 0

                                for i in acroMapStates.values():
                                    if i in aff:
                                        row.append(i)
                                        states = 1
                                        break

                                if states == 0:
                                    row.append('-')

                                if 'IG' in aff or 'Inspector General' in aff or 'Inspec. Ge
                                    row.append('Yes')
                                else:
                                    row.append('No')

                            else:
                                row.append('-')
                                row.append('-')
                                row.append('-')
                                row.append('-')
                                row.append('-')
                                row.append('-')

                            all.append(row)

                        #except:
                        #    writer.writerows(all)
                        #    continue
                        writer.writerows(all)
```

In [ ]:

```
from fuzzywuzzy import fuzz
from fuzzywuzzy import process
print (fuzz.partial_ratio( 'Hon. Peter J. Visclosky, a Representative in Congress from the
```

In [ ]:

```
# GPO agencies for metadata
# Exact matching on agency names and acronyms, states, Inspector General

import re
import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math
import csv
import pandas as pd
gpo = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_server
gpo2 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
results_csvs_new1 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearin


sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

sample500 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_

sample500GPOOutput = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_heari

df1 = pd.read_csv(sample500)
#print(df1['filename'])

df = pd.read_csv(gpo2)
agencies = []
for i in (df['Agency']):
    temp = i.replace('U.S.', 'United States')
    temp = temp.replace('US', 'United States')
    temp = temp.replace('Dep.', 'Department')
    temp = temp.replace('Dept.', 'Department')
    temp = temp.replace('Dept', 'Department')
    temp = temp.replace('Assoc', 'Association')
    temp = temp.replace('Assoc.', 'Association')
    temp = temp.replace('Brd', 'Board')
    temp = temp.replace('Brd.', 'Board')
    temp = temp.replace('DC', 'District of Columbia')
    temp = temp.replace('D.C.', 'District of Columbia')

    temp = temp.replace('.,',' ')
    temp = temp.replace('.;',' ')
    temp = temp.replace('.-',' ')
    temp = temp.replace('.:',' ')
    temp = temp.replace('.,',' ')

    temp = temp.replace('.', '')

    for i in temp.split():
        if i in acroMap.keys():
            temp = temp.replace(i,acroMap[i])

    for i in temp.split():
        if i in acroMapStates.keys():
            temp = temp.replace(i,acroMapStates[i])

    agencies.append(temp)

JK = []
UA = []
Parent = []

for i in (df['JK Code']):
    JK.append(i)
for i in (df['UA Code']):
    UA.append(i)
for i in (df['Parent UA Code']):
    Parent.append(i)


#print (set(agencies))

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#file = pd.read_csv(sample_csvs + 'CHRG-105hhrg40051' +'.csv')

#for file in df1['filename']:

#    try:

                #print ( row1['Affiliation'] + ' : ' + agency + '\t' + str(max_score))
#agencies = agencies[:100]
#for file in set(os.listdir(results_csvs)):
#    if file not in set(os.listdir(results_csvs_new)):
with open(metadata_results,'r', encoding="utf8") as csvinput:
            with open(metadata_results_new, 'w+', encoding="utf8") as csvoutput:
                        writer = csv.writer(csvoutput, lineterminator='\n')
                        reader = csv.reader(csvinput)

                        all = []
                        row = next(reader)

                        row.append('Agency')
                        row.append('JK code')
                        row.append('UA code')
                        row.append('Parent UA code')
                        row.append('US State')
                        row.append('Inspector General')
                        all.append(row)
                        #print (row)
                        #try:

                        for row in reader:

                            if row[13].strip() != '-':
                                max_score = 0
                                agency = '-'
                                jk = '-'
                                ua = '-'
                                parent = '-'

                                agencyL = []
                                jkL = []
                                uaL = []
                                parentL = []
                                stateL = []
                                IGL = []

                                if row[13] == 'Refer column R':
                                    affs = row[17].split('\n')
                                else:
                                    affs = row[13].split('\n')

                                for aff in affs:
                                    if aff.strip() != '':
                                        aff = aff.replace('U.S.', 'United States')
                                        aff = aff.replace('US', 'United States')
                                        aff = aff.replace('Dep.', 'Department')
                                        aff = aff.replace('Dept.', 'Department')
                                        aff = aff.replace('Dept', 'Department')
                                        aff = aff.replace('Assoc', 'Association')
                                        aff = aff.replace('Assoc.', 'Association')
                                        aff = aff.replace('Brd', 'Board')
                                        aff = aff.replace('Brd.', 'Board')
                                        aff = aff.replace('DC', 'District of Columbia')
                                        aff = aff.replace('D.C.', 'District of Columbia')

                                        aff = aff.replace('.,',' ')
                                        aff = aff.replace('.;',' ')
                                        aff = aff.replace('.-',' ')
                                        aff = aff.replace('.:',' ')
                                        aff = aff.replace('.,',' ')

                                        aff = aff.replace('.', '')

                                        for i in aff.split():
                                            if i in acroMap.keys():
                                                aff = aff.replace(i,acroMap[i])

                                        for i in aff.split():
                                            if i in acroMapStates.keys():
                                                aff = aff.replace(i,acroMapStates[i])
```

```
                                        hit = 0
                                        for i in ((agencies)):
                                            #score = fuzz.WRatio( i, aff )
                                            #if (score > max_score):
                                            if i in aff:
                                                #max_score = score
                                                agency = i
                                                index = agencies.index(i)
                                                jk = JK[index]
                                                ua = UA[index]
                                                parent = Parent[index]

                                                agencyL.append(str(agency))
                                                jkL.append(str(jk))
                                                uaL.append(str(ua))
                                                parentL.append(str(parent))
                                                hit = 1
                                                break
                                        '''
                                        if max_score >= 90:
                                            row.append(agency)
                                            row.append(jk)
                                            row.append(ua)
                                            row.append(parent)

                                        else:
                                            row.append('-')
                                            row.append('-')
                                            row.append('-')
                                            row.append('-')
                                        '''

                                        if hit == 0:
                                            agencyL.append('-')
                                            jkL.append('-')
                                            uaL.append('-')
                                            parentL.append('-')

                                        states = 0

                                        for i in acroMapStates.values():
                                            if i in aff:
                                                stateL.append(i)
                                                states = 1
                                                break

                                        if states == 0:
                                            stateL.append('-')

                                        if 'IG' in aff or 'Inspector General' in aff or 'Ir
                                            IGL.append('Yes')
                                        else:
                                            IGL.append('No')

                                row.append("\n".join(agencyL))
                                row.append("\n".join(jkL))
                                row.append("\n".join(uaL))
                                row.append("\n".join(parentL))
                                row.append("\n".join(stateL))
                                row.append("\n".join(IGL))
```

```
        else:
            row.append('-')
            row.append('-')
            row.append('-')
            row.append('-')
            row.append('-')
            row.append('-')

        all.append(row)

    #except:
    #    writer.writerows(all)
    #    continue
    writer.writerows(all)
```

In [ ]:

```
# Adding "Bills" column in all individual CSVs

import re
import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

FullTexts = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_


#df1 = pd.read_csv(metadata_results)
#print(df1['filename'])

sample_jackets = ['CHRG-115hhrg27211']
count = 0

#files = set(os.listdir(results_csvs)) - set(os.listdir(results_csvs_new))

countWitness = 0
```

```
gpo = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_server
gpo2 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

sample500 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_

sample500GPOOutput = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_heari

sample500SAOutput = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearin

sampleBill = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from

#df1 = pd.read_csv(sample500)
#print(df1['filename'])

#sid = SentimentIntensityAnalyzer()

#print (set(agencies))

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#file = pd.read_csv(sample_csvs + 'CHRG-105hhrg40051' +'.csv')
count = 0
#for file in set(os.listdir(results_csvs)):
for file in set(os.listdir(results_csvs)):
    #print (file)
    #print (set(os.listdir(results_csvs)))
    #file = file + '.csv'
    #if file in set(os.listdir(results_csvs)):
    file = file.replace('.csv','')


    with open(results_csvs+file+'.csv','r', encoding="utf8") as csvinput:
            with open(results_csvs_new+file+'.csv', 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    row.append('Bills')
                    all.append(row)


                    for row in reader:
                        if re.search(r"(S\.\d{4})",row[12]) or re.search(r"(S\. \d{4})",row
                        #if re.search(r"(.)*(S\.\d{4})*(S\. \d{4})*(S\d{4})*(S \d{4})*(H\.R
                            row.append('1')
                            count += 1
                           # print(count)
                        else:
                            row.append('0')
                        all.append(row)
                        #break
                    writer.writerows(all)
```

In [ ]:

```
print ("No. of bills found : ")
print (count)
```

In [39]:

```
import requests
import os
import json
import xmltodict

import csv
import pandas as pd

months = ['01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12']
years = ['1995', '1996', '1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004', '2
committees = [102, 104, 106, 113, 115, 124, 128, 134, 138, 142, 156, 164, 173, 176, 182, 18
congresses = [104, 105, 106, 107, 108, 109, 110, 111, 112]

gpoShort = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_s

df = pd.read_csv(gpoShort)
agencies = []

for i in (df['Agency']):
    agencies.append(i)

JK = []
UA = []
Parent = []

for i in (df['JK Code']):
    JK.append(i)
for i in (df['UA Code']):
    UA.append(i)
for i in (df['Parent UA Code']):
    Parent.append(i)
```

In [40]:

```
# CSV 1: Number of utterances made by the agency about a bill per month

CSV1 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

#for file in set(os.listdir(results_csvs)):
    #print (file)
    #print (set(os.listdir(results_csvs)))
    #file = file + '.csv'
    #if file in set(os.listdir(results_csvs)):
#file = file.replace('.csv','')


#with open(results_csvs+file+'.csv','r', encoding="utf8") as csvinput:
with open(CSV1, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    writer.writerow(["Date", "Committee", "Agency", "JK Code", "UA Code", "

                    for committee in committees:
                        for month in months:
                            for year in years:
                                for i in range(len(agencies)):
                                    row_temp = "=\"" +month+'-'+year+"\"", committee, agenc
                                    writer.writerow(row_temp)
```

In [ ]:

```

```

In [37]:

```
# To remove duplicate ent
CSV1 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve
from more_itertools import unique_everseen
with open(CSV1,'r') as f, open('2.csv','w') as out_file:
    out_file.writelines(unique_everseen(f))
```

In [ ]:

```
print(len(CSV1Dict.keys()))
print(list(CSV1Dict.values())[0:100000])
print(CSV1Dict['06-1998 344 United States Postal Service'])
```

In [ ]:

```
# CSV 1: Number of utterances made by the agency about a bill per month

CSV221 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_ser

monthDict ={'01':'Jan',
           '02':'Feb',
            '03':'Mar',
            '04':'Apr',
            '05':'May',
            '06':'Jun',
            '07':'Jul',
            '08':'Aug',
            '09':'Sep',
            '10':'Oct',
            '11':'Nov',
            '12':'Dec'

           }
with open(CSV1,'r', encoding="utf8") as csvinput:
            with open(CSV221, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    row.append('Number of utterances made by the agency about a bill per mo
                    all.append(row)

                    for row in reader:
                        CSV1RowDate = str(row[0])
                        CSV1RowDate = CSV1RowDate.replace('=', '')
                        CSV1RowDate = CSV1RowDate.replace('"', '')

                        CSV1key =  CSV1RowDate+' '+ row[1] +' '+ row[2].strip()

                        if CSV1key in CSV1Dict.keys():
                            row.append(CSV1Dict[CSV1key])

                        all.append(row)
                        #break
                    writer.writerows(all)
```

In [ ]:

```
# Number of utterances made by the agency per month - CSV2


import re
import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd


CSV1 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve
results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr

CSV1Dict = {}

utteranceCount = []

with open(CSV1, 'r', encoding="utf8") as csvinput2:
    CSV1reader = csv.reader(csvinput2)

    #all = []
    CSV1row = next(CSV1reader)

    #CSV1row.append('Number of utterances made by the agency about a bill per month')
    #all.append(CSV1row)

    for CSV1row in CSV1reader:
        count = 0

        CSV1RowDate = str(CSV1row[0])
        CSV1RowDate = CSV1RowDate.replace('=', '')
        CSV1RowDate = CSV1RowDate.replace('"', '')

        CSV1key =  CSV1RowDate+' '+ CSV1row[1] +' '+ CSV1row[2].strip()

        CSV1Dict[CSV1key.strip()] = 0

print(len(CSV1Dict.keys()))
print(list(CSV1Dict.keys())[0])

for file in set(os.listdir(results_csvs)):

            file = file.replace('.csv','')

            with open(results_csvs+file+'.csv','r', encoding="utf8") as csvinput:
                   # with open(CSV1, 'w+', encoding="utf8") as csvoutput:
                      #      writer = csv.writer(csvoutput, lineterminator='\n')
                            reader = csv.reader(csvinput)
                            row = next(reader)
                            for row in reader:
                                #if row[27] == '1':

                                    date = row[13].split('-')[0]+'-'+row[13].split('-')[2]
                                    indCSVkey = date +' '+ row[0] + ' '+ row[21].strip()

                                    #print(indCSVkey)

                                    if indCSVkey.strip() in CSV1Dict.keys():
                                        CSV1Dict[indCSVkey.strip()] += 1
                                        #print(indCSVkey)

        #print(count)
        #utteranceCount.append(count)
```

In [ ]:

```
# Number of utterances made by the agency per month - CSV2

CSV211 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_ser

monthDict ={'01':'Jan',
           '02':'Feb',
            '03':'Mar',
            '04':'Apr',
            '05':'May',
            '06':'Jun',
            '07':'Jul',
            '08':'Aug',
            '09':'Sep',
            '10':'Oct',
            '11':'Nov',
            '12':'Dec'

           }

with open(CSV1,'r', encoding="utf8") as csvinput:
            with open(CSV211, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    row.append('Number of utterances made by the agency per month')
                    all.append(row)

                    for row in reader:
                        CSV1RowDate = str(row[0])
                        CSV1RowDate = CSV1RowDate.replace('=', '')
                        CSV1RowDate = CSV1RowDate.replace('"', '')

                        CSV1key =  CSV1RowDate+' '+ row[1] +' '+ row[2].strip()

                        if CSV1key in CSV1Dict.keys():
                            row.append(CSV1Dict[CSV1key])

                        all.append(row)
                        #break
                    writer.writerows(all)
```

In [ ]:

```
# For each committee, need the number of total utterances per month - CSV3


import re
import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd


CSV1 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve
results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr

CSV1Dict = {}

utteranceCount = []

with open(CSV1, 'r', encoding="utf8") as csvinput2:
    CSV1reader = csv.reader(csvinput2)

    #all = []
    CSV1row = next(CSV1reader)

    #CSV1row.append('Number of utterances made by the agency about a bill per month')
    #all.append(CSV1row)

    for CSV1row in CSV1reader:
        count = 0

        CSV1RowDate = str(CSV1row[0])
        CSV1RowDate = CSV1RowDate.replace('=', '')
        CSV1RowDate = CSV1RowDate.replace('"', '')

        CSV1key =  CSV1RowDate+' '+ CSV1row[1] #+' '+ CSV1row[2].strip()

        CSV1Dict[CSV1key.strip()] = 0

print(len(CSV1Dict.keys()))
print(list(CSV1Dict.keys())[0])

for file in set(os.listdir(results_csvs)):

            file = file.replace('.csv','')

            with open(results_csvs+file+'.csv','r', encoding="utf8") as csvinput:
                   # with open(CSV1, 'w+', encoding="utf8") as csvoutput:
                      #      writer = csv.writer(csvoutput, lineterminator='\n')
                            reader = csv.reader(csvinput)
                            row = next(reader)
                            for row in reader:
                                #if row[27] == '1':

                                    date = row[13].split('-')[0]+'-'+row[13].split('-')[2]
                                    indCSVkey = date +' '+ row[0] #+ ' '+ row[21].strip()

                                    #print(indCSVkey)

                                    if indCSVkey.strip() in CSV1Dict.keys():
                                        CSV1Dict[indCSVkey.strip()] += 1
                                        #print(indCSVkey)

        #print(count)
        #utteranceCount.append(count)
```

In [ ]:

```
# For each committee, need the number of total utterances per month - CSV3


CSV2111 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_se

monthDict ={'01':'Jan',
           '02':'Feb',
            '03':'Mar',
            '04':'Apr',
            '05':'May',
            '06':'Jun',
            '07':'Jul',
            '08':'Aug',
            '09':'Sep',
            '10':'Oct',
            '11':'Nov',
            '12':'Dec'

           }

with open(CSV1,'r', encoding="utf8") as csvinput:
            with open(CSV2111, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    row.append('Number of utterances made by the committees per month')
                    all.append(row)

                    for row in reader:
                        CSV1RowDate = str(row[0])
                        CSV1RowDate = CSV1RowDate.replace('=', '')
                        CSV1RowDate = CSV1RowDate.replace('"', '')

                        CSV1key =  CSV1RowDate+' '+ row[1] #+' '+ row[2].strip()

                        if CSV1key in CSV1Dict.keys():
                            row.append(CSV1Dict[CSV1key])

                        all.append(row)
                        #break
                    writer.writerows(all)
```

In [ ]:

```
# For each agency, need the number of total utterances per month - CSV4


import re
import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd


CSV1 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve
results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr

CSV1Dict = {}

utteranceCount = []

with open(CSV1, 'r', encoding="utf8") as csvinput2:
    CSV1reader = csv.reader(csvinput2)

    #all = []
    CSV1row = next(CSV1reader)

    #CSV1row.append('Number of utterances made by the agency about a bill per month')
    #all.append(CSV1row)

    for CSV1row in CSV1reader:
        count = 0

        CSV1RowDate = str(CSV1row[0])
        CSV1RowDate = CSV1RowDate.replace('=', '')
        CSV1RowDate = CSV1RowDate.replace('"', '')

        CSV1key =  CSV1RowDate+' '+ CSV1row[1].strip()

        CSV1Dict[CSV1key.strip()] = 0

print(len(CSV1Dict.keys()))
print(list(CSV1Dict.keys())[0])

for file in set(os.listdir(results_csvs)):

            file = file.replace('.csv','')

            with open(results_csvs+file+'.csv','r', encoding="utf8") as csvinput:
                   # with open(CSV1, 'w+', encoding="utf8") as csvoutput:
                      #      writer = csv.writer(csvoutput, lineterminator='\n')
                            reader = csv.reader(csvinput)
                            row = next(reader)
                            for row in reader:
                                #if row[27] == '1':

                                    date = row[13].split('-')[0]+'-'+row[13].split('-')[2]
                                    indCSVkey = date + ' '+ row[21].strip()

                                    #print(indCSVkey)

                                    if indCSVkey.strip() in CSV1Dict.keys():
                                        CSV1Dict[indCSVkey.strip()] += 1
                                        #print(indCSVkey)

        #print(count)
        #utteranceCount.append(count)
```

In [ ]:

```
# For each agency, need the number of total utterances per month - CSV4



CSV4 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

monthDict ={'01':'Jan',
           '02':'Feb',
            '03':'Mar',
            '04':'Apr',
            '05':'May',
            '06':'Jun',
            '07':'Jul',
            '08':'Aug',
            '09':'Sep',
            '10':'Oct',
            '11':'Nov',
            '12':'Dec'

           }

with open(CSV1,'r', encoding="utf8") as csvinput:
            with open(CSV4, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    row.append('Number of utterances made by the agencies per month')
                    all.append(row)

                    for row in reader:
                        CSV1RowDate = str(row[0])
                        CSV1RowDate = CSV1RowDate.replace('=', '')
                        CSV1RowDate = CSV1RowDate.replace('"', '')

                        CSV1key =  CSV1RowDate+' '+ row[1].strip()

                        if CSV1key in CSV1Dict.keys():
                            row.append(CSV1Dict[CSV1key])

                        all.append(row)
                        #break
                    writer.writerows(all)
```

In [ ]:

```
# Number of hearings made by the agency per month - CSV5


import re
import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd


CSV1 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve
results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr

CSV1Dict = {}

utteranceCount = []

with open(CSV1, 'r', encoding="utf8") as csvinput2:
    CSV1reader = csv.reader(csvinput2)

    #all = []
    CSV1row = next(CSV1reader)

    #CSV1row.append('Number of utterances made by the agency about a bill per month')
    #all.append(CSV1row)

    for CSV1row in CSV1reader:
        count = 0

        CSV1RowDate = str(CSV1row[0])
        CSV1RowDate = CSV1RowDate.replace('=', '')
        CSV1RowDate = CSV1RowDate.replace('"', '')

        CSV1key =  CSV1RowDate+' '+ CSV1row[1] +' '+ CSV1row[2].strip()

        CSV1Dict[CSV1key.strip()] = 0

print(len(CSV1Dict.keys()))
print(list(CSV1Dict.keys())[0])


hearingsSet = set()
for file in set(os.listdir(results_csvs)):

            file = file.replace('.csv','')

            hearingsSet.clear()
            with open(results_csvs+file+'.csv','r', encoding="utf8") as csvinput:
                   # with open(CSV1, 'w+', encoding="utf8") as csvoutput:
                      #      writer = csv.writer(csvoutput, lineterminator='\n')
                            reader = csv.reader(csvinput)
                            row = next(reader)
                            for row in reader:
                                #if row[27] == '1':

                                    date = row[13].split('-')[0]+'-'+row[13].split('-')[2]
                                    indCSVkey = date +' '+ row[0] + ' '+ row[21].strip()

                                    hearingsSet.add(indCSVkey)
                                    #print(indCSVkey)

            for i in hearingsSet:
                if i.strip() in CSV1Dict.keys():
                    CSV1Dict[i.strip()] += 1
                                        #print(indCSVkey)

        #print(count)
        #utteranceCount.append(count)


print(len(CSV1Dict.keys()))
print(list(CSV1Dict.values())[0])
```

In [ ]:

```
# Number of hearings made by the agency per month - CSV5

CSV5 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

monthDict ={'01':'Jan',
           '02':'Feb',
            '03':'Mar',
            '04':'Apr',
            '05':'May',
            '06':'Jun',
            '07':'Jul',
            '08':'Aug',
            '09':'Sep',
            '10':'Oct',
            '11':'Nov',
            '12':'Dec'

           }

with open(CSV1,'r', encoding="utf8") as csvinput:
            with open(CSV5, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    row.append('Number of hearings made by the agency per month')
                    all.append(row)

                    for row in reader:
                        CSV1RowDate = str(row[0])
                        CSV1RowDate = CSV1RowDate.replace('=', '')
                        CSV1RowDate = CSV1RowDate.replace('"', '')

                        CSV1key =  CSV1RowDate+' '+ row[1] +' '+ row[2].strip()

                        if CSV1key in CSV1Dict.keys():
                            row.append(CSV1Dict[CSV1key])

                        all.append(row)
                        #break
                    writer.writerows(all)
```

In [ ]:

```
# Finding gender based on names

import re
import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd


CSV1 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve
results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr

namesDict = {}

for file in set(os.listdir(results_csvs)):

            file = file.replace('.csv','')

            hearingsSet.clear()
            with open(results_csvs+file+'.csv','r', encoding="utf8") as csvinput:
                   # with open(CSV1, 'w+', encoding="utf8") as csvoutput:
                      #      writer = csv.writer(csvoutput, lineterminator='\n')
                            reader = csv.reader(csvinput)
                            row = next(reader)
                            for row in reader:
                                if row[17].strip() != 'NA' or row[17].strip() != '-':
                                    namesDict[row[17]] = 'M/F'
                                    #print(indCSVkey)
```

```
print(len(namesDict.keys()))
print(list(namesDict.values())[0])
```

In [ ]:

```
# Finding gender based on names

import gender_guesser.detector as gender
d = gender.Detector()

for i in range(30):
    print((list(namesDict.keys())[i]) + " : "+ d.get_gender(list(namesDict.keys())[i]))
    #print('\n')

print(d.get_gender(u"Mainzer"))
print(d.get_gender(u"Bob"))
```

In [ ]:

```
import nltk
nltk.download('names')
```

In [ ]:

```
# Finding gender based on names

import random
from nltk.corpus import names
import nltk

def gender_features(word):
    return {'last_letter':word[-1]}

# preparing a list of examples and corresponding class labels.
labeled_names = ([(name, 'male') for name in names.words('male.txt')]+
             [(name, 'female') for name in names.words('female.txt')])

random.shuffle(labeled_names)

# we use the feature extractor to process the names data.
featuresets = [(gender_features(n), gender)
               for (n, gender)in labeled_names]

# Divide the resulting list of feature
# sets into a training set and a test set.
train_set, test_set = featuresets[5:], featuresets[:5]

# The training set is used to
# train a new "naive Bayes" classifier.
classifier = nltk.NaiveBayesClassifier.train(train_set)


for i in range(30):
    print((list(namesDict.keys())[i]) + " : "+ classifier.classify(gender_features((list(na
    #print('\n')

print(classifier.classify(gender_features('Bob')))
```

In [ ]:

```

print(len(namesDict.keys()))
print(list(namesDict.values())[0])
```

In [ ]:

```
fout = "namesDict.txt"
fo = open(fout, "w")

for k, v in namesDict.items():
    fo.write(str(k) +'\n')

fo.close()
```

In [ ]:

```
count = 0

for k, v in namesDict.items():
    if (str(k).find(',')!=-1 ):
        count += 1

print (count)
```

In [ ]:

```
# Metadata
# subCommittee extraction

import json
from pprint import pprint

import sys
import csv

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

APIs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve


df1 = pd.read_csv(metadata_results)
#print(df1['filename'])

sample_jackets = ['CHRG-115hhrg27211']
```

```
count = 0

with open(metadata_results,'r', encoding="utf8") as csvinput:
                with open(metadata_results_new, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    row.append('subCommittee')
                    all.append(row)


                    for row in reader:
                        try:

                            file = row[6] + ".json"

                            with open(APIs+file) as data_file:
                                jsonObj = json.load(data_file)



                                if (jsonObj["mods"]["extension"][2]["congCommittee"]["subCo
                                    subCommittee = jsonObj["mods"]["extension"][2]["congCom
                                    row.append(subCommittee)
                                    #print (subCommittee)
                                else:
                                    row.append('-')

                        except:
                            row.append("-")

                        all.append(row)

                    #except:
                    #    writer.writerows(all)
                    #    continue
                    writer.writerows(all)
```

In [ ]:

```
# Metadata
# Column: "Committee member count"

import json
from pprint import pprint

import sys
import csv

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

APIs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve


df1 = pd.read_csv(metadata_results)
#print(df1['filename'])

sample_jackets = ['CHRG-115hhrg27211']
```

```
count = 0

with open(metadata_results,'r', encoding="utf8") as csvinput:
                with open(metadata_results_new, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    row.append('Committee member count')
                    all.append(row)


                    for row in reader:
                            count = len(row[14].split('\n'))

                            row.append(count)

                            all.append(row)

                    #except:
                    #    writer.writerows(all)
                    #    continue
                    writer.writerows(all)
```

In [ ]:

```
# Metadata
# Column: "Denominator count"

import json
from pprint import pprint

import sys
import csv

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

APIs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

House = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serv
Senate = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_ser

CongCom = {}
```

```
with open(House,'r', encoding="utf8") as csvinput:
                   # with open(CSV1, 'w+', encoding="utf8") as csvoutput:
                      #      writer = csv.writer(csvoutput, lineterminator='\n')
                            reader = csv.reader(csvinput)
                            row = next(reader)
                            for row in reader:
                                if(row[0]+':'+row[1] in CongCom.keys()):
                                    CongCom[row[0]+':'+row[1]] += 1
                                else:
                                    CongCom[row[0]+':'+row[1]] = 1

with open(Senate,'r', encoding="utf8") as csvinput:
                   # with open(CSV1, 'w+', encoding="utf8") as csvoutput:
                      #      writer = csv.writer(csvoutput, lineterminator='\n')
                            reader = csv.reader(csvinput)
                            row = next(reader)
                            for row in reader:
                                if(row[0]+':'+row[1] in CongCom.keys()):
                                    CongCom[row[0]+':'+row[1]] += 1
                                else:
                                    CongCom[row[0]+':'+row[1]] = 1



#print(CongCom)

with open(metadata_results,'r', encoding="utf8") as csvinput:
                with open(metadata_results_new, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    row.append('Denominator count')
                    all.append(row)


                    for row in reader:
                            if (row[2].replace("th","")+':'+row[3]) in CongCom.keys():
                                count = CongCom[(row[2].replace("th","")+':'+row[3])]
                            else:
                                count = '-'

                            row.append(count)
                            all.append(row)

                    #except:
                    #    writer.writerows(all)
                    #    continue
                    writer.writerows(all)
```

In [ ]:

```
# Metadata
# Column: "Party count"

import json
from pprint import pprint

import sys
import csv

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_frc
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

APIs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

House = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serv
Senate = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_ser

CongCom = {}
```

```
with open(House,'r', encoding="utf8") as csvinput:
                   # with open(CSV1, 'w+', encoding="utf8") as csvoutput:
                      #      writer = csv.writer(csvoutput, lineterminator='\n')
                            reader = csv.reader(csvinput)
                            row = next(reader)
                            for row in reader:
                                if(row[0]+':'+row[1]+':'+row[6] in CongCom.keys()):
                                    CongCom[row[0]+':'+row[1]+':'+row[6]] += 1
                                else:
                                    CongCom[row[0]+':'+row[1]+':'+row[6]] = 1

with open(Senate,'r', encoding="utf8") as csvinput:
                   # with open(CSV1, 'w+', encoding="utf8") as csvoutput:
                      #      writer = csv.writer(csvoutput, lineterminator='\n')
                            reader = csv.reader(csvinput)
                            row = next(reader)
                            for row in reader:
                                if(row[0]+':'+row[1]+':'+row[6] in CongCom.keys()):
                                    CongCom[row[0]+':'+row[1]+':'+row[6]] += 1
                                else:
                                    CongCom[row[0]+':'+row[1]+':'+row[6]] = 1



#print(CongCom)

with open(metadata_results,'r', encoding="utf8") as csvinput:
                with open(metadata_results_new, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    row.append('Party count(100:200:328:999:9999)')
                    all.append(row)


                    for row in reader:
                            if (row[2].replace("th","")+':'+row[3]+':100') in CongCom.keys(
                                count100 = CongCom[(row[2].replace("th","")+':'+row[3]+':10
                            else:
                                count100 = '-'

                            if (row[2].replace("th","")+':'+row[3]+':200') in CongCom.keys(
                                count200 = CongCom[(row[2].replace("th","")+':'+row[3]+':20
                            else:
                                count200 = '-'

                            if (row[2].replace("th","")+':'+row[3]+':328') in CongCom.keys(
                                count328 = CongCom[(row[2].replace("th","")+':'+row[3]+':32
                            else:
                                count328 = '-'

                            if (row[2].replace("th","")+':'+row[3]+':999') in CongCom.keys(
                                count999 = CongCom[(row[2].replace("th","")+':'+row[3]+':99
                            else:
                                count999 = '-'

                            if (row[2].replace("th","")+':'+row[3]+':9999') in CongCom.keys
                                count9999 = CongCom[(row[2].replace("th","")+':'+row[3]+':9
                            else:
                                count9999 = '-'

                            temp = "=\"" + str(count100) + ":" + str(count200) +  ":" + str

                            row.append( temp)
                            all.append(row)

                    #except:
                    #    writer.writerows(all)
                    #    continue
                    writer.writerows(all)
```

In [ ]:

```
#Finding unique party codes
PartyCodes = {}

with open(House,'r', encoding="utf8") as csvinput:
                   # with open(CSV1, 'w+', encoding="utf8") as csvoutput:
                      #      writer = csv.writer(csvoutput, lineterminator='\n')
                            reader = csv.reader(csvinput)
                            row = next(reader)
                            for row in reader:
                                if(row[6] in PartyCodes.keys()):
                                    PartyCodes[row[6]] += 1
                                else:
                                    PartyCodes[row[6]] = 0

print(PartyCodes.keys())
```

In [ ]:

```
# Metadata
# Column: "Party & Committee info:"

import json
from pprint import pprint

import sys
import csv

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

APIs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

House = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serv
Senate = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_ser

PartyCom = {}
```

```
with open(House,'r', encoding="utf8") as csvinput:
                   # with open(CSV1, 'w+', encoding="utf8") as csvoutput:
                      #      writer = csv.writer(csvoutput, lineterminator='\n')
                            reader = csv.reader(csvinput)
                            row = next(reader)
                            for row in reader:
                                PartyCom[row[0]+row[1]+row[3].lower().strip()] = row[6]+':'


with open(Senate,'r', encoding="utf8") as csvinput:
                   # with open(CSV1, 'w+', encoding="utf8") as csvoutput:
                      #      writer = csv.writer(csvoutput, lineterminator='\n')
                            reader = csv.reader(csvinput)
                            row = next(reader)
                            for row in reader:
                                PartyCom[row[0]+row[1]+row[3].lower().strip()] = row[6]+':'


#print(CongCom)

with open(metadata_results,'r', encoding="utf8") as csvinput:
                with open(metadata_results_new, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    row.append('Party & Committee info(Party:Senior Party Member:Committee
                    all.append(row)

                    for row in reader:
                            temp = []
                            for name in row[14].split('\n'):
                                if row[2].replace("th","")+row[3]+name.split(' : ')[0].lowe
                                    temp.append(  PartyCom[row[2].replace("th","")+row[3]+r
                                else:
                                    temp.append( '-'+'-'+'-')

                            row.append("\n".join(temp))
                            all.append(row)

                    #except:
                    #    writer.writerows(all)
                    #    continue
                    writer.writerows(all)
```

In [ ]:

```
# Metadata
# Column: "Expertise"

import json
from pprint import pprint

import sys
import csv

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

APIs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

House = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serv
Senate = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_ser

expertise={
    'A.A.' : 'Associate of Arts',
        'A.S.' : 'Associate of Science',
        'A.A.S.' : 'Associate of Applied Science',
        'ADN' : 'Associates Degree in Nursing',
        'B.A.' : 'Bachelor of Arts',
        'B.S.' : 'Bachelor of Science',
        'B.E.' : 'Bachelor of Engineering',
        'M.A.' : 'Master of Arts',
        'M.S.' : 'Master of Science',
        'MBA' : 'Master of Business Administration',
        'M.Ed.' : 'Master of Education',
        'Ph.D.' : 'Doctor of Philosophy',
        'DNP' : 'Doctor of Nursing Practice',
        'Ed.D.' : 'Doctor of Education',
         'J.D.' : 'Juris Doctorate, a law degree',
        'M.D.' : 'Medical Doctor, a physicians degree',
        'D.D.S.' : 'Doctor of Dental Surgery, a dentistry degree',
'Pharm.D.' : 'Doctor of Pharmacy , a pharmaceutical medicine degree'


}

with open(metadata_results,'r', encoding="utf8") as csvinput:
                with open(metadata_results_new, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    #row.append('Expertise')
                    all.append(row)

                    for row in reader:
                            temp = []
                            if row[13].strip() != '-':

                                if row[13].strip() == 'Refer column R':
                                    affs = row[17].split('\n')
                                else:
                                    affs = row[13].split('\n')

                                for name in affs:
                                    if name.strip() != '':
                                        done = 0
                                        for i in name.split():
                                            #print (i)
                                            if i.strip() in expertise.keys():
                                                temp.append(i+' : '+expertise[i])
                                                done = 1
                                                break
                                               # print(i)

                                        if done == 0:
                                                temp.append('-')

                            else:
                                temp.append('-')

                            row[29] = ("\n".join(temp))
                            all.append(row)
                            #break

                    #except:
                    #    writer.writerows(all)
                    #    continue
                    writer.writerows(all)
```

In [ ]:

```
# Metadata
# GPO Plumbook
# Column: "Type of Appt., Title"

import json
from pprint import pprint

from collections import defaultdict

import sys
import csv

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_frc
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

APIs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

House = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serv
Senate = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_ser
```

```
GPO = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_server

GPODict = defaultdict(list)

files = set(os.listdir(GPO))

for file in files:
    #file=file.replace('.csv','')
    with open(GPO+file,'r', encoding="utf8") as csvinput:
                   # with open(CSV1, 'w+', encoding="utf8") as csvoutput:
                      #      writer = csv.writer(csvoutput, lineterminator='\n')
                            reader = csv.reader(csvinput)
                            row = next(reader)
                            for row in reader:
                                GPODict[row[0].lower().strip()].append(row[7].lower().strip

print(list(GPODict.keys())[0:15])

print(list(GPODict.values())[0:15])


with open(metadata_results,'r', encoding="utf8") as csvinput:
                with open(metadata_results_new, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    row.append('From GPO Plumbook')
                    all.append(row)

                    for row in reader:
                            temp = []

                            for agency in row[18].split('\n'):
                                if agency.lower().strip() in GPODict.keys():
                                    if row[13].strip() != "Refer column R":
                                        index = row[18].split('\n').index(agency)
                                        #for witness in row[13].split('\n'):
                                        witness = row[13].split('\n')[index]
                                        for item in GPODict[agency.lower().strip()]:
                                                done = 0
                                                name = item.split(' :: ')[0].lower().strip(
                                                if fuzz.token_sort_ratio(( witness.lower().
                                                    appt = item.split(' :: ')[2].strip()
                                                    title = item.split(' :: ')[1].strip()
                                                    temp.append(title +' :: '+appt)
                                                    done = 1
                                                    break
                                        if done == 0:
                                                    temp.append('-')
                                    else:
                                        index = row[18].split('\n').index(agency)
                                        #for witness in row[13].split('\n'):
                                        witness = row[17].split('\n')[index]
                                        for item in GPODict[agency.lower().strip()]:
                                                done = 0
                                                name = item.split(' :: ')[0].lower().strip(
                                                if fuzz.token_sort_ratio(( witness.lower().
                                                    appt = item.split(' :: ')[2].strip()
                                                    title = item.split(' :: ')[1].strip()
                                                    temp.append(title +' :: '+appt)
                                                    done = 1
                                                    break
                                        if done == 0:
                                                    temp.append('-')

                                else:
                                    temp.append('-')

                            row.append("\n".join(temp))
                            all.append(row)

                            #break

                    #except:
                    #    writer.writerows(all)
                    #    continue
                    writer.writerows(all)
```

In [ ]:

```
# Gender, to find unique names
# Individual CSVs

namesDict = {}

for file in set(os.listdir(results_csvs)):

            file = file.replace('.csv','')

            with open(results_csvs+file+'.csv','r', encoding="utf8") as csvinput:
                   # with open(CSV1, 'w+', encoding="utf8") as csvoutput:
                      #      writer = csv.writer(csvoutput, lineterminator='\n')
                            reader = csv.reader(csvinput)
                            row = next(reader)
                            for row in reader:
                                if row[17].strip() != 'NA' or row[17].strip() != '-':
                                    namesDict[row[17]] = 'M/F'
                                    #print(indCSVkey)



print(len(namesDict.keys()))
print(list(namesDict.values())[0])

count = 0

fout = "namesDict(Mem+Wit).txt"
fo = open(fout, "w")

for k, v in namesDict.items():
    if (str(k).find(',')!=-1 ):
        count += 1
    else:
        fo.write(str(k) +'\n')

fo.close()

print (count)
print (len(namesDict.keys()) - count )
```

In [ ]:

```
# Meta metadata
# Witness level info.:


import json
from pprint import pprint

from collections import defaultdict

import sys
import csv

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

APIs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

House = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serv
Senate = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_ser
```

```
GPO = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_server



MetaMetadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

with open(metadata_results,'r', encoding="utf8") as csvinput:
                with open(MetaMetadata_results, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)
                    temp = []
                    temp.append('Filename')
                    temp.append('Witnesses')
                    temp.append('Scrapped witnesses')
                    temp.append('Agency')
                    temp.append('JK code')
                    temp.append('UA code')
                    temp.append('Parent UA code')
                    temp.append('US State')
                    temp.append('Inspector General')
                    temp.append('Expertise')
                    temp.append('From GPO Plumbook(Title :: Appt)')

                    all.append(temp)

                    for row in reader:

                            if row[13].strip() != '-':

                                if row[13].strip() == 'Refer column R':
                                    affs = row[17].split('\n')
                                else:
                                    affs = row[13].split('\n')
                                #all = []
                                for aff in affs:
                                    temp = []

                                    if aff.strip() != '':
                                        try:
                                            temp.append(row[6])
                                            index = affs.index(aff)

                                            if row[13].strip() == 'Refer column R':
                                                temp.append('-')
                                                temp.append(row[17].split('\n')[index])

                                            else:
                                                temp.append(row[13].split('\n')[index])
                                                temp.append('-')


                                            temp.append(row[18].split('\n')[index])
                                            temp.append(row[19].split('\n')[index])
                                            temp.append(row[20].split('\n')[index])
                                            temp.append(row[21].split('\n')[index])
                                            temp.append(row[22].split('\n')[index])
                                            temp.append(row[23].split('\n')[index])
                                            temp.append(row[29].split('\n')[index])
                                            temp.append(row[30].split('\n')[index])


                                            all.append(temp)

                                        except:
                                            print (row)
                                #writer.writerows(all)
                            #all.append(row)
                            #break

                    #except:
                    #    writer.writerows(all)
                    #    continue
                    writer.writerows(all)
```

In [ ]:

```
# Metadata
# Column: Keywords
# keywords "oversight," "investigation," or "budget request"

import json
from pprint import pprint

from collections import defaultdict

import sys
import csv

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

APIs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

House = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serv
Senate = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_ser
```

```
GPO = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_server

keyWordsDict = {'oversight' : 1,
               'investigation' : 1,
               'budget request' : 1}

with open(metadata_results, 'r', encoding="utf8") as csvinput:
    with open(metadata_results_new, 'w+', encoding="utf8") as csvoutput:
            reader = csv.reader(csvinput)
            writer = csv.writer(csvoutput, lineterminator='\n')


            row = next(reader)

            all = []
            row.append('Keywords present')
            all.append(row)

            for row in reader:

                            file = row[6].strip()+'.csv'
                            cleaned = ''

                            if row[15] == 'Yes':
                                with open(results_csvs+file,'r', encoding="utf8") as csvinp
                                                reader1 = csv.reader(csvinput1)
                                                row1 = next(reader1)
                                                for row1 in reader1:
                                                    cleaned += row1[12]


                            done = 0
                            for i in cleaned.split():
                                    if str(i).lower().strip() in keyWordsDict.keys():
                                        row.append('Yes')
                                        done = 1
                                        break
                            if done == 0:
                                row.append('No')

                            all.append(row)


            writer.writerows(all)
```

In [3]:

```
# Commitee: Agency : Congress   -> Triplet
# CSV1


import json
from pprint import pprint

from collections import defaultdict

import sys
import csv

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing



CSV1 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve

tripletDict = {}

with open(metadata_results,'r', encoding="utf8") as csvinput:
                   # with open(CSV1,http://localhost:8888/notebooks/Documents/RA%20NLP/RA_N
                      #      writer = csv.writer(csvoutput, lineterminator='\n')
                            reader = csv.reader(csvinput)
                            row = next(reader)
                            for row in reader:
                                    triplet = row[2].strip().replace('th','')+' :: '+row[3]
                                    for agency in row[18].split('\n'):
                                        if agency.strip()!='' and agency.strip()!='-':
                                            triplet += ' :: '+agency
                                            if triplet.strip() in tripletDict.keys():
                                                tripletDict[triplet.strip()] += 1
                                            else:
                                                tripletDict[triplet.strip()] = 1

                                        triplet = row[2].strip().replace('th','')+' :: '+ro

print(len(tripletDict.keys()))

print(len(tripletDict.values()))

print(list(tripletDict.keys())[0:10])

with open(CSV1, 'w+', encoding="utf8") as csvoutput:
            writer = csv.writer(csvoutput, lineterminator='\n')

            row = []
            all = []
            row.append('Congress')
            row.append('Commitee')
            row.append('Agency')
            row.append('Count')
            all.append(row)

            for key in tripletDict.keys():
                            row = []
                            item = key.split(' :: ')
                            row.append(item[0])
                            row.append(item[1])
                            row.append(item[2])
                            row.append(tripletDict[key])
                            all.append(row)



            writer.writerows(all)
```

```
5874
5874
['115 :: 102 :: Federal Reserve', '115 :: 102 :: Department of Agriculture',
'115 :: 102 :: Farm Credit Administration', '115 :: 102 :: Department of the
Treasury', '115 :: 104 :: Office of Community Planning and Development', '11
5 :: 104 :: Federal Highway Administration', '115 :: 106 :: Central Intellig
ence Agency', '115 :: 106 :: Department of Defense', '115 :: 173 :: Departme
nt of Transportation', '115 :: 106 :: Joint Chiefs of Staff']
```

In [5]:

```
# Metadata
# Columns:
#           Attendance proportion %
#           Bills
#           subpoena

import json
from pprint import pprint

from collections import defaultdict

import sys
import csv

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd

metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing
metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea

results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
results_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing

sample_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fro
sample_csvs_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings

APIs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve
```

```
House = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serv
Senate = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_ser


GPO = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_server



with open(metadata_results, 'r', encoding="utf8") as csvinput:
    with open(metadata_results_new, 'w+', encoding="utf8") as csvoutput:
            reader = csv.reader(csvinput)
            writer = csv.writer(csvoutput, lineterminator='\n')


            row = next(reader)

            all = []

            row.append('Attendance proportion %')
            row.append('Bills')
            row.append('subpoena')

            all.append(row)

            for row in reader:

                            file = row[6].strip()+'.csv'


                            try:
                                row.append((int(row[25])/int(row[26])) *100)
                            except:
                                row.append('-')

                            cleaned = ''
                            bills = 0
                            if row[15] == 'Yes':
                                with open(results_csvs+file,'r', encoding="utf8") as csvinp
                                                reader1 = csv.reader(csvinput1)
                                                row1 = next(reader1)
                                                bills = 0
                                                for row1 in reader1:
                                                    cleaned += row1[12]
                                                    if str(row1[-1]).strip() == '1':
                                                        bills = 1

                            row.append(bills)

                            done = 0
                            subpoena = 0

                            for i in cleaned.split():
                                    if 'subpoena' == str(i).lower().strip() :
                                        subpoena += 1
                                        done = 1

                            if done == 0:
                                row.append('No')

                            else:
                                row.append(subpoena)
                            all.append(row)

            writer.writerows(all)
```

In [34]:

```
# CSV 1: At the hearing level
# Comm. - Agency - Month


import re
import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd


CSV1 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve
results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing


CSV1Dict = {}
Attendance = {}

utteranceCount = []

with open(CSV1, 'r', encoding="utf8") as csvinput2:
    CSV1reader = csv.reader(csvinput2)

    #all = []
    CSV1row = next(CSV1reader)

    #CSV1row.append('Number of utterances made by the agency about a bill per month')
    #all.append(CSV1row)
    for CSV1row in CSV1reader:
        count = 0

        CSV1RowDate = str(CSV1row[0])
        CSV1RowDate = CSV1RowDate.replace('=', '')
        CSV1RowDate = CSV1RowDate.replace('"', '')

        CSV1key =  CSV1RowDate+' '+ CSV1row[1] +' '+ CSV1row[2].strip()

        CSV1Dict[str(CSV1key.strip())] = 0

        Attendance[str(CSV1key.strip())]  = 0

print(len(CSV1Dict.keys()))
print(list(CSV1Dict.keys())[0])

with open(metadata_results,'r', encoding="utf8") as csvinput:
                                reader = csv.reader(csvinput)
                                row = next(reader)
                                for row in reader:
                                        if row[16]!='-' and row[-3]!='-' and str(row[-12].s
                                            if ';' in row[16].strip():
                                                date = row[16].split(';')[0]
                                                date = date.split('-')[1]+'-'+date.split('-
                                            else:
                                                date = row[16].split('-')[1]+'-'+row[16].sp

                                            for agency in row[18].split('\n'):

                                                if agency != '':
                                                    indCSVkey = date +' '+ row[3].strip() +

                                                    if indCSVkey.strip() in CSV1Dict.keys()
                                                        CSV1Dict[str(indCSVkey.strip())] +=
                                                        Attendance[str(indCSVkey.strip())]



print(len(CSV1Dict.keys()))
print(list(CSV1Dict.keys())[0:10])
```

```
593352
01-1995 102 Broadcasting Board of Governors
593352
['01-1995 102 Broadcasting Board of Governors', '01-1995 102 Commission on C
ivil Rights', '01-1995 102 Commodities Futures Trading Commission', '01-1995
102 Consumer Product Safety Commission', '01-1995 102 Court Services and Off
ender Supervision Agency', '01-1995 102 Department of Agriculture', '01-1995
102 Department of Commerce', '01-1995 102 Department of Defense', '01-1995 1
02 Department of Education', '01-1995 102 Department of Energy']
```

In [35]:

```
# CSV 1: At the hearing level
# Comm. - Agency - Month

CSV221 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_ser

monthDict ={'01':'Jan',
           '02':'Feb',
            '03':'Mar',
            '04':'Apr',
            '05':'May',
            '06':'Jun',
            '07':'Jul',
            '08':'Aug',
            '09':'Sep',
            '10':'Oct',
            '11':'Nov',
            '12':'Dec'

           }
with open(CSV1,'r', encoding="utf8") as csvinput:
            with open(CSV221, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    #row.append('Count of all hearings')
                    #row.append('Count that requires keywords')
                    #row.append('Count of only non-bill hearings')
                    #row.append('Count of only non-bill hearings that require keywords')
                    #row.append('Average # of committee members attending for all hearings'
                    #row.append('Average # of committee members attending for all hearings
                    #row.append('Average # of committee members attending for non-bill hear
                    #row.append('Average # of committee members attending for non-bill hear
                    #row.append('Average # of committee members attending for all hearings
                    row.append('Average # of committee members attending for non-bill heari



                    all.append(row)

                    for row in reader:
                        CSV1RowDate = str(row[0])
                        CSV1RowDate = CSV1RowDate.replace('=', '')
                        CSV1RowDate = CSV1RowDate.replace('"', '')

                        CSV1key =  CSV1RowDate+' '+ row[1] +' '+ row[2].strip()

                        try:
                            row.append(float(Attendance[str(CSV1key)] / float(CSV1Dict[str(
                        except:
                            row.append('-')

                        #row.append(CSV1Dict[str(CSV1key)])

                        all.append(row)
                        writer.writerows(all)
```

In [ ]:

```
# CSV 1: At the hearing level
# Comm. - Agency - Month


import re
import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd


CSV1 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve
results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing


CSV1Dict = {}
Attendance = {}

utteranceCount = []

with open(CSV1, 'r', encoding="utf8") as csvinput2:
    CSV1reader = csv.reader(csvinput2)

    #all = []
    CSV1row = next(CSV1reader)

    #CSV1row.append('Number of utterances made by the agency about a bill per month')
    #all.append(CSV1row)
    for CSV1row in CSV1reader:
        count = 0

        CSV1RowDate = str(CSV1row[0])
        CSV1RowDate = CSV1RowDate.replace('=', '')
        CSV1RowDate = CSV1RowDate.replace('"', '')

        CSV1key =  CSV1RowDate+' '+ CSV1row[1] +' '+ CSV1row[2].strip()

        CSV1Dict[str(CSV1key.strip())] = 0

        Attendance[str(CSV1key.strip())]  = 0

print(len(CSV1Dict.keys()))
print(list(CSV1Dict.keys())[0])

with open(metadata_results,'r', encoding="utf8") as csvinput:
                                reader = csv.reader(csvinput)
                                row = next(reader)
                                for row in reader:
                                        if row[16]!='-' and row[-3]!='-' and str(row[-12].s
                                            if ';' in row[16].strip():
                                                date = row[16].split(';')[0]
                                                date = date.split('-')[1]+'-'+date.split('-
                                            else:
                                                date = row[16].split('-')[1]+'-'+row[16].sp

                                            for agency in row[18].split('\n'):

                                                if agency != '':
                                                    indCSVkey = date +' '+ row[3].strip() +

                                                    if indCSVkey.strip() in CSV1Dict.keys()
                                                        CSV1Dict[str(indCSVkey.strip())] +=
                                                        Attendance[str(indCSVkey.strip())]



print(len(CSV1Dict.keys()))
print(list(CSV1Dict.keys())[0:10])
```

In [21]:

```
# CSV 1: At the utterance level
# Comm. - Agency - Month


import re
import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd


CSV1 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve
results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing


CSV1Dict = {}
Attendance = {}

utteranceCount = []

with open(CSV1, 'r', encoding="utf8") as csvinput2:
    CSV1reader = csv.reader(csvinput2)

    #all = []
    CSV1row = next(CSV1reader)

    #CSV1row.append('Number of utterances made by the agency about a bill per month')
    #all.append(CSV1row)
    for CSV1row in CSV1reader:
        count = 0

        CSV1RowDate = str(CSV1row[0])
        CSV1RowDate = CSV1RowDate.replace('=', '')
        CSV1RowDate = CSV1RowDate.replace('"', '')

        CSV1key =  CSV1RowDate+' '+ CSV1row[1] +' '+ CSV1row[2].strip()

        CSV1Dict[str(CSV1key.strip())] = 0

        Attendance[str(CSV1key.strip())]  = 0

print(len(CSV1Dict.keys()))
print(list(CSV1Dict.keys())[0])

df1 = pd.read_csv(metadata_results)
for file in set(os.listdir(results_csvs)):

            file = file.replace('.csv','')

            index = df1['Filename'].tolist().index(file)
            bills = df1['Bills'].tolist()[index]
            keywords = df1['Keywords present'].tolist()[index]
            attendance = df1['Attendance proportion %'].tolist()[index]
            subCommittee = str(df1['subCommittee'].tolist()[index])
            IG = str(df1['Inspector General'].tolist()[index])

            if IG=='Yes' and str(bills).strip() == '0':#and subCommittee == '-':# and str(k

                with open(results_csvs+file+'.csv','r', encoding="utf8") as csvinput:
                       # with open(CSV1, 'w+', encoding="utf8") as csvoutput:
                          #      writer = csv.writer(csvoutput, lineterminator='\n')
                                reader = csv.reader(csvinput)
                                row = next(reader)
                                for row in reader:

                                        if row[13]!='-' and row[-7]!='-':# and str(row[-12]

                                                    date = row[13].split('-')
                                                    date = date[0]+'-'+date[2]
                                                    indCSVkey = date +' '+ row[0].strip() +

                                                    if indCSVkey.strip() in CSV1Dict.keys()
                                                            CSV1Dict[str(indCSVkey.strip())
                                                            Attendance[str(indCSVkey.strip(



print(len(CSV1Dict.keys()))
print(list(CSV1Dict.keys())[0:10])

print(list(CSV1Dict.values())[0:10])
```

```
593352
01-1995 102 Broadcasting Board of Governors

C:\Users\RAHUL\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:
2785: DtypeWarning: Columns (10) have mixed types. Specify dtype option on i
```

```
mport or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)


593352
['01-1995 102 Broadcasting Board of Governors', '01-1995 102 Commission on C
ivil Rights', '01-1995 102 Commodities Futures Trading Commission', '01-1995
102 Consumer Product Safety Commission', '01-1995 102 Court Services and Off
ender Supervision Agency', '01-1995 102 Department of Agriculture', '01-1995
102 Department of Commerce', '01-1995 102 Department of Defense', '01-1995 1
02 Department of Education', '01-1995 102 Department of Energy']
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

In [22]:

```
# CSV 1: At the hearing level
# Comm. - Agency - Month

CSV221 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_ser

monthDict ={'01':'Jan',
           '02':'Feb',
            '03':'Mar',
            '04':'Apr',
            '05':'May',
            '06':'Jun',
            '07':'Jul',
            '08':'Aug',
            '09':'Sep',
            '10':'Oct',
            '11':'Nov',
            '12':'Dec'

           }
with open(CSV1,'r', encoding="utf8") as csvinput:
            with open(CSV221, 'w+', encoding="utf8") as csvoutput:
                    writer = csv.writer(csvoutput, lineterminator='\n')
                    reader = csv.reader(csvinput)

                    all = []
                    row = next(reader)

                    #row.append('Count of all utterances')
                    #row.append('Count of all utterances that require keywords')
                    #row.append('Count of all utterances for only non-bill hearings')
                    #row.append('Count of all utterances for only non-bill hearings that re
                    #row.append('Average # of committee members attending for all hearings
                    #row.append('Average # of committee members attending for all hearings
                    #row.append('Average # of committee members attending for non-bill hear
                    #row.append('Average # of committee members attending for non-bill hear
                    #row.append('Average # of committee members attending for all hearings
                    row.append('Average # of committee members attending for non-bill heari

                    all.append(row)

                    for row in reader:
                        CSV1RowDate = str(row[0])
                        CSV1RowDate = CSV1RowDate.replace('=', '')
                        CSV1RowDate = CSV1RowDate.replace('"', '')

                        CSV1key =  CSV1RowDate+' '+ row[1] +' '+ row[2].strip()

                        try:
                            row.append(float(Attendance[str(CSV1key)] / float(CSV1Dict[str(
                        except:
                            row.append('-')

                        #row.append(CSV1Dict[str(CSV1key)])

                        all.append(row)

                    writer.writerows(all)
```

In [2]:

```
# MetaMetadata_results : cleaning column B


import re
import sys
import csv

#csv.field_size_limit(sys.maxsize)

maxInt = sys.maxsize
decrement = True

while decrement:
    # decrease the maxInt value by factor 10
    # as long as the OverflowError occurs.

    decrement = False
    try:
        csv.field_size_limit(maxInt)
    except OverflowError:
        maxInt = int(maxInt/10)
        decrement = True

import os

import math

import requests
import xml.etree.ElementTree as ET

import json
import xmltodict

import csv
import pandas as pd


CSV1 = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_from_serve
results_csvs = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearings_fr
MetaMetadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea
MetaMetadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional


CSV1Dict = {}
Attendance = {}

utteranceCount = []

expertise={

    'A.A.' : 'Associate of Arts',
        'A.S.' : 'Associate of Science',
        'A.A.S.' : 'Associate of Applied Science',
        'ADN' : 'Associates Degree in Nursing',
        'B.A.' : 'Bachelor of Arts',
        'B.S.' : 'Bachelor of Science',
        'B.E.' : 'Bachelor of Engineering',
        'M.A.' : 'Master of Arts',
        'M.S.' : 'Master of Science',
        'MBA' : 'Master of Business Administration',
        'M.Ed.' : 'Master of Education',
        'Ph.D.' : 'Doctor of Philosophy',
        'DNP' : 'Doctor of Nursing Practice',
        'Ed.D.' : 'Doctor of Education',
         'J.D.' : 'Juris Doctorate, a law degree',
        'M.D.' : 'Medical Doctor, a physicians degree',
        'D.D.S.' : 'Doctor of Dental Surgery, a dentistry degree',
'Pharm.D.' : 'Doctor of Pharmacy , a pharmaceutical medicine degree'


}


with open(MetaMetadata_results, 'r', encoding="utf8") as csvinput:
    with open(MetaMetadata_results_new, 'w+', encoding="utf8") as csvoutput:
            reader = csv.reader(csvinput)
            writer = csv.writer(csvoutput, lineterminator='\n')


            all = []

            row = next(reader)
            all.append(row)

            for row in reader:

                            row[1]=row[1].split(':')[0]

                            for word in row[1].split():
                                if word in expertise.keys():
                                    row[1] = row[1].replace(word, '')

                            all.append(row)


            writer.writerows(all)
```

In [6]:

```


metadata_results_new = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hea
metadata_results = "D:/USC/RA NLP/Hearing data/congressional_hearings/congressional_hearing


with open(metadata_results, 'r', encoding="utf8") as csvinput:
    with open(metadata_results_new, 'w+', encoding="utf8") as csvoutput:
            reader = csv.reader(csvinput)
            writer = csv.writer(csvoutput, lineterminator='\n')


            all = []

            row = next(reader)
            all.append(row)

            for row in reader:

                            if row[-11].strip() != '-':
                                row[-3] = '-'

                            all.append(row)


            writer.writerows(all)
```

In [ ]: