q1-wordsimilarity-l215625

March 11, 2024

[ ]:
```
import numpy as np
import pandas as pd
from itertools import combinations
```

The problem is to match the user's free-form input against a pre-determined list of banks. For example, user input 'bawag bank' should be matched to 'BAWAG Group AG'.

[ ]:
```
# List of banks to compare
banks =   ['Sberbank Europe AG',
           'BAWAG Group AG',
           'Raiffeisenbankengruppe OÖ Verbund eGen',
           'Raiffeisen Bank International AG',
           'Volksbanken Verbund',
           'Erste Group Bank AG',
           'KBC Groep',
           'Investeringsmaatschappij Argenta',
           'Belfius Bank',
           'AXA Bank Belgium',
           'The Bank of New York Mellon SA/NV',
           'First Investment Bank AD',
           'RCB Bank Ltd',
           'Bank of Cyprus Holdings Public Limited Company',
           'Hellenic Bank Public Company Limited',
           'DekaBank Deutsche Girozentrale',
           'Erwerbsgesellschaft der S-Finanzgruppe mbH & Co. KG',
           'UBS Europe SE',
           'DEUTSCHE APOTHEKER- UND ÄRZTEBANK EG',
           'Volkswagen Bank Gesellschaft mit beschränkter Haftung',
           'Münchener Hypothekenbank eG',
           'DZ BANK AG Deutsche Zentral-Genossenschaftsbank, Frankfurt am Main',
           'HASPA Finanzholding',
           'State Street Europe Holdings Germany S.a.r.l. & Co. KG',
           'J.P. Morgan AG',
           'DEUTSCHE BANK AKTIENGESELLSCHAFT',
           'COMMERZBANK Aktiengesellschaft',
           'Landesbank Baden-Württemberg',
           'Landesbank Hessen-Thüringen Girozentrale',
           'Norddeutsche Landesbank - Girozentrale -',
```

```
        'Deutsche Pfandbriefbank AG',
        'Aareal Bank AG',
        'Hamburg Commercial Bank AG',
        'Bayerische Landesbank',
        'Jyske Bank A/S',
        'Sydbank A/S',
        'Nykredit Realkredit A/S',
        'Danske Bank A/S',
        'Luminor Holding AS',
        'Abanca Corporacion Bancaria S.A.',
        'Banco Santander S.A.',
        'Ibercaja Banco S.A.',
        'Kutxabank S.A',
        'Unicaja Banco S.A.',
        'CaixaBank S.A.',
        'Banco de Crédito Social Cooperativo',
        'Banco Bilbao Vizcaya Argentaria S.A.',
        'Banco de Sabadell S.A.',
        'Bankinter S.A.',
        'Kuntarahoitus Oyj',
        'Nordea Bank Abp',
        'OP Osuuskunta',
        'SFIL',
        'RCI Banque',
        'Confédération Nationale du Crédit Mutuel',
        'La Banque Postale',
        'Bpifrance',
        "C.R.H. - Caisse de refinancement de l'habitat",
        'HSBC Continental Europe',
        'Groupe BPCE',
        'Groupe Crédit Agricole',
        'Société générale',
        'BNP Paribas',
        'ALPHA SERVICES AND HOLDINGS S.A.',
        'National Bank of Greece S.A.',
        'Eurobank Ergasias Services and Holdings S.A.',
        'Piraeus Financial Holdings',
        'OTP-csoport',
        'Magyar Bankholding',
        'Barclays Bank Ireland plc',
        'Citibank Holdings Ireland Limited',
        'AIB Group plc',
        'Bank of Ireland Group plc',
        'Ulster Bank Ireland Designated Activity Company',
        'Bank of America Europe Designated Activity Company',
        'Íslandsbanki hf.',
        'Landsbankinn hf.',
```

```
        'Arion banki hf',
        'Intesa Sanpaolo S.p.A.',
        'Gruppo Bancario Finecobank  ',
        'UniCredit S.p.A.',
        'Gruppo Bancario Mediolanum  ',
        'Credito Emiliano Holding S.p.A.',
        'Banco BPM SpA',
        'Banca Popolare di Sondrio, Società Cooperativa per Azioni',
        'Banca Monte dei Paschi di Siena S.p.A.',
        'CASSA CENTRALE BANCA',
        'ICCREA BANCA S.P.A.',
        'Mediobanca - Banca di Credito Finanziario S.p.A.',
        'Akcine bendrove Šiauliu bankas',
        'Precision Capital S.A.',
        'RBC Investor Services Bank S.A.',
        'J.P. Morgan Bank Luxembourg S.A.',
        'Banque Internationale à Luxembourg',
        'Banque et Caisse d´Epargne de l´Etat, Luxembourg',
        'Akciju sabiedriba "Citadele banka"',
        'MDB Group Limited',
        'Bank of Valletta Plc',
        'HSBC Bank Malta p.l.c.',
        'BNG Bank N.V.',
        'ING Groep N.V.',
        'LP Group B.V.',
        'de Volksbank N.V.',
        'ABN AMRO Bank N.V.',
        'Coöperatieve Rabobank U.A.',
        'Nederlandse Waterschapsbank N.V.',
        'Bank Polska Kasa Opieki S.A.',
        'Powszechna Kasa Oszczednosci Bank Polski S.A.',
        'LSF Nani Investments S.à r.l.',
        'Banco Comercial Português SA',
        'Caixa Geral de Depósitos SA',
        'Banca Transilvania',
        'Länförsäkringar Bank AB (publ)',
        'Kommuninvest - group',
        'Skandinaviska Enskilda Banken - group',
        'SBAB Bank AB - group',
        'Swedbank - group',
        'Svenska Handelsbanken - group',
        'Biser Topco S.à r.l.',
        'Nova Ljubljanska Banka d.d. Ljubljana']
```

[ ]:
```
# Examples of search strings
s1 = 'Bawag bank' # other options: 'Bawag bank', 'Erste', 'Raiffaisen bank'
```

[ ]:
```
# A naive search method which you need to improve
from difflib import SequenceMatcher

res = []
for token in banks:
  res.append([s1, token, SequenceMatcher(None, s1, token).ratio()])

df2 = pd.DataFrame(res, columns=['Bank 1', 'Bank 2', 'Score'])
# The outcome is not great, for this search query 'BAWAG Group AG' should have␣
 ↪highest similarity
df2.sort_values(by=['Score'], ascending=[False]).head()
```

[ ]:

	Bank 1	Bank 2	Score
8	Bawag bank	Belfius Bank	0.454545
12	Bawag bank	RCB Bank Ltd	0.454545
33	Bawag bank	Bayerische Landesbank	0.451613
42	Bawag bank	Kutxabank S.A	0.434783
99	Bawag bank	BNG Bank N.V.	0.434783

[ ]:
```
#The desired combination has a low score
idx = df2['Bank 2'].isin(['BAWAG Group AG'])

df2[idx].sort_values(by=['Score'], ascending=[False]).head()
```

[ ]:

	Bank 1	Bank 2	Score
1	Bawag bank	BAWAG Group AG	0.166667

[ ]:
```
s1 = 'Bawag bank' # other options: 'Bawag bank', 'Erste', 'Raiffaisen bank'
```

[ ]:

	Bank 1	Bank 2	Score
99	Bawag bank	BNG Bank N.V.	0.625000
31	Bawag bank	Aareal Bank AG	0.555556
116	Bawag bank	Swedbank - group	0.428571
0	Bawag bank	Sberbank Europe AG	0.416667
115	Bawag bank	SBAB Bank AB - group	0.416667

[ ]:
```
idx = df2['Bank 2'].isin(['BAWAG Group AG'])
df2[idx].sort_values(by=['Score'], ascending=[False]).head()
```

[ ]:

	Bank 1	Bank 2	Score
1	Raiffeisen bank	BAWAG Group AG	0.214286

[ ]:
```
def similarity(s1,s2):

  s1=s1.lower().split(' ')
  s2=s2.lower().split(' ')
```

```
  s1 = [i for i in s1 if i != 'bank']
  s2 = [i for i in s2 if i != 'bank']

  s1=''.join([i for i in s1 if i!=' '])
  s2=''.join([i for i in s2 if i!=' '])


  set1=set(s1)
  set2=set(s2)


  intersect_length=0
  union=set1.union(set2)
  set1=[int(i in set1) for i in union]
  set2=[int(i in set2) for i in union]
  for i in range(0,len(set1)):
    if set1[i]==set2[i]:
      intersect_length+=1
  return round(abs(intersect_length)/abs(len(union)),2)

s1 = 'Bawag bank'
res = []
for token in banks:
  res.append([s1, token, similarity(s1, token)])

df2 = pd.DataFrame(res, columns=['Bank 1', 'Bank 2', 'Score'])
# The outcome is not great, for this search query 'BAWAG Group AG' should have␣
 ↪highest similarity
df2.sort_values(by=['Score'], ascending=[False])
```

[ ]:

	Bank 1	Bank 2	Score
1	Bawag bank	BAWAG Group AG	0.50
115	Bawag bank	SBAB Bank AB - group	0.30
9	Bawag bank	AXA Bank Belgium	0.30
31	Bawag bank	Aareal Bank AG	0.29
99	Bawag bank	BNG Bank N.V.	0.29
..	…	…	…
108	Bawag bank	LSF Nani Investments S.à r.l.	0.06
113	Bawag bank	Kommuninvest - group	0.06
54	Bawag bank	Confédération Nationale du Crédit Mutuel	0.06
52	Bawag bank	SFIL	0.00
67	Bawag bank	OTP-csoport	0.00

[120 rows x 3 columns]

[ ]:
```
# prompt: now give a test code to test the above similarity function
```

```python
# Test the similarity function
s1 = "Bawag bank"
s2 = "BAWAG Group AG"
similarity = sim(s1, s2)
print(f"Similarity between '{s1}' and '{s2}': {similarity}")

s1 = "Erste"
s2 = "Erste Group Bank AG"
similarity = sim(s1, s2)
print(f"Similarity between '{s1}' and '{s2}': {similarity}")

s1 = "Raiffeisen bank"
s2 = "Raiffeisenbankengruppe OÖ Verbund eGen"
similarity = sim(s1, s2)
print(f"Similarity between '{s1}' and '{s2}': {similarity}")
```

```
{'a': 1, 'g': 1, 'p': 0, 'b': 1, 'w': 1, 'u': 0, 'o': 0, 'r': 0, 'k': 1, 'n': 1}
{'a': 1, 'g': 1, 'p': 1, 'b': 1, 'w': 1, 'u': 1, 'o': 1, 'r': 1, 'k': 0, 'n': 0}
Similarity between 'Bawag bank' and 'BAWAG Group AG': 0.3999999999999999
{'a': 0, 'g': 0, 'e': 1, 's': 1, 'p': 0, 'n': 0, 'b': 0, 'u': 0, 'r': 1, 't': 1,
'k': 0, 'o': 0}
{'a': 1, 'g': 1, 'e': 1, 's': 1, 'p': 1, 'n': 1, 'b': 1, 'u': 1, 'r': 1, 't': 1,
'k': 1, 'o': 1}
Similarity between 'Erste' and 'Erste Group Bank AG': 0.33333333333333337
{'a': 1, 'g': 0, 'e': 1, 'v': 0, 's': 1, 'p': 0, 'ö': 0, 'd': 0, 'f': 1, 'b': 1,
'u': 0, 'o': 0, 'r': 1, 'k': 1, 'i': 1, 'n': 1}
{'a': 1, 'g': 1, 'e': 1, 'v': 1, 's': 1, 'p': 1, 'ö': 1, 'd': 1, 'f': 1, 'b': 1,
'u': 1, 'o': 1, 'r': 1, 'k': 1, 'i': 1, 'n': 1}
Similarity between 'Raiffeisen bank' and 'Raiffeisenbankengruppe OÖ Verbund
eGen': 0.5625
```

[ ]: