```python
import numpy as np
import pandas as pd
from itertools import combinations
from difflib import SequenceMatcher
from nltk.stem import WordNetLemmatizer
import string
import pandas as pd
from unidecode import unidecode
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

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'.

```python
# 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
```

Use Normalization techniques Unicode,threshold

```
def normalize_text(text):
    normalized_text = unidecode(text)
    translator = str.maketrans("", "", string.punctuation)
    normalized_text = normalized_text.lower().translate(translator)
    return normalized_text
similarity_threshold = 0.5

res = []
normalized_s1 = normalize_text(s1)

for token in banks:
    normalized_token = normalize_text(token)
    seq_matcher = SequenceMatcher(None, normalized_s1,
normalized_token)
    similarity_score = seq_matcher.ratio()

    if similarity_score >= similarity_threshold:
        res.append([s1, token, similarity_score])

df2 = pd.DataFrame(res, columns=['Bank 1', 'Bank 2', 'Score'])
df2.sort_values(by=['Score'], ascending=[False]).head()

        Bank 1          Bank 2     Score
10  Bawag bank   BNG Bank N.V.  0.666667
0   Bawag bank  BAWAG Group AG  0.583333
4   Bawag bank  Aareal Bank AG  0.583333
```

```
1   Bawag bank    Belfius Bank  0.545455
3   Bawag bank    RCB Bank Ltd  0.545455
```

```
idx = df2['Bank 2'].isin(['BAWAG Group AG'])

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

```
       Bank 1          Bank 2     Score
0  Bawag bank  BAWAG Group AG  0.583333
```

Use TFIDF

```
all_banks = [s1] + banks
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(all_banks)
cosine_similarities = cosine_similarity(tfidf_matrix[0],
tfidf_matrix[1:])[0]
res = pd.DataFrame({'Bank 1': [s1] * len(banks), 'Bank 2': banks,
'Cosine Similarity': cosine_similarities})
df_sorted = res.sort_values(by='Cosine Similarity', ascending=False)
df_sorted.head()
```

```
        Bank 1           Bank 2  Cosine Similarity
1   Bawag bank   BAWAG Group AG           0.628456
34  Bawag bank   Jyske Bank A/S           0.177352
8   Bawag bank     Belfius Bank           0.177352
99  Bawag bank    BNG Bank N.V.           0.177352
37  Bawag bank  Danske Bank A/S           0.177352
```

```
idx = df_sorted['Bank 2'].isin(['BAWAG Group AG'])
filtered_results = df_sorted[idx].sort_values(by='Cosine Similarity',
ascending=False)
filtered_results.head()
```

```
       Bank 1          Bank 2  Cosine Similarity
1  Bawag bank  BAWAG Group AG           0.628456
```