AI for Legal Assistance

April 13, 2021

1 AILA: Artificial Intelligence for Legal Assisatance

1.1 Similar Case Matching

1.1.1 We are given a dataset consisting of 2914 prior cases and a test dataset of 50 queries. We need to retrieve the most similar prior case for each of the queries.

[ ]:
```
# This Python 3 environment comes with many helpful analytics libraries␣
↪installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/
↪docker-python

#Imports
import glob
import functools
import datetime as dt
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
import random
import re
import numpy as np
import pandas as pd

# Input data files are available in the read-only "../input/" directory

import os
#for dirname, _, filenames in os.walk('/kaggle/input/legalai/Object_casedocs'):
#    for filename in filenames:
#        print(os.path.join(dirname, filename))
# the above code will list all files under the input directory
```

1.2 Data Handling

1.2.1 We first wrap up all the text files into a single csv(comma separated file)

[2]:
```
import glob
import csv

read_files = glob.glob('/kaggle/input/legalai/Object_casedocs/*')

with open("object_casedocs.csv", "w") as outfile:
    w=csv.writer(outfile)
    for f in read_files:
        with open(f, "r") as infile:
            w.writerow([" ".join([line.strip() for line in infile])])
```

1.2.2 A Glimpse about how the data inside the csv file looks!

[3]:
```
df = pd.read_csv('object_casedocs.csv',header=None)
df.columns = ["Text"]
df
```

[3]:
```
                                                    Text
0     L. Laxmikanta v State by Superintendent of Pol…
1     Homi Rajvansh v State of Maharashtra and other…
2     Direct Recruit Class Ii Engineering OfficersAs…
3     Rajinder Kumar Kindra v Delhi Administration T…
4     Kalyan and Others v State of Uttar Pradesh Sup…
…                                                  …
2909  Haryana State Cooperative Labour and others v …
2910  State of Karnataka v Chikkahottappa Alias Vara…
2911  Kilari Malakondiaah @ Malayadri and Others v S…
2912  Kanthimathy Plantations Pvt- Limited v State O…
2913  Union of India and Others v K. P. Prabhakaran …

[2914 rows x 1 columns]
```

1.2.3 Let us get some basic information about the data

[4]:
```
len(df)
```

[4]: 2914

[5]:
```
df.shape
```

[5]: (2914, 1)

[6]:
```
df.info
```

[6]:
```
<bound method DataFrame.info of
Text
0     L. Laxmikanta v State by Superintendent of Pol…
1     Homi Rajvansh v State of Maharashtra and other…
2     Direct Recruit Class Ii Engineering OfficersAs…
3     Rajinder Kumar Kindra v Delhi Administration T…
4     Kalyan and Others v State of Uttar Pradesh Sup…
…                                                  …
2909  Haryana State Cooperative Labour and others v …
2910  State of Karnataka v Chikkahottappa Alias Vara…
2911  Kilari Malakondiaah @ Malayadri and Others v S…
2912  Kanthimathy Plantations Pvt- Limited v State O…
2913  Union of India and Others v K. P. Prabhakaran …

[2914 rows x 1 columns]>
```

1.3 Text preprocessing techniques: Cleansing the data

1.3.1 1. Convert to lowercase, remove punctuation and special characters, using RegeX and strip

1.3.2 2. Remove stopwords

1.3.3 3. Stemming

1.3.4 4. Lemmatization

[7]:
```
import re
#Convert lowercase remove punctuation and Character and then strip
text = df.iloc[0]
print(text)
text = re.sub(r'[^\w\s]', '', str(text).lower().strip())
txt = text.split()
print(txt)
```

```
Text    L. Laxmikanta v State by Superintendent of Pol…
Name: 0, dtype: object
['text', 'l', 'laxmikanta', 'v', 'state', 'by', 'superintendent', 'of', 'pol',
'name', '0', 'dtype', 'object']
```

[8]:
```
#remove stopwords
import nltk
```

```
lst_stopwords = nltk.corpus.stopwords.words("english")
txt = [word for word in txt if word not in lst_stopwords]
print(txt)
```

```
['text', 'l', 'laxmikanta', 'v', 'state', 'superintendent', 'pol', 'name', '0',
'dtype', 'object']
```

[9]:
```
#stemming
ps = nltk.stem.porter.PorterStemmer()
print([ps.stem(word) for word in txt])
```

```
['text', 'l', 'laxmikanta', 'v', 'state', 'superintend', 'pol', 'name', '0',
'dtype', 'object']
```

[10]:
```
#Lemmetization
nltk.download('wordnet')
lem = nltk.stem.wordnet.WordNetLemmatizer()
print([lem.lemmatize(word) for word in txt])
```

```
[nltk_data] Downloading package wordnet to /usr/share/nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
['text', 'l', 'laxmikanta', 'v', 'state', 'superintendent', 'pol', 'name', '0',
'dtype', 'object']
```

1.4 Preprocessing the data: Apply these techniques on all records of the dataset

[11]:
```
#to apply all the technique to all the records on dataset
def utils_preprocess_text(text, flg_stemm=True, flg_lemm =True,␣
↪lst_stopwords=None ):
    text = re.sub(r'[^\w\s]', '', str(text).lower().strip())

    #tokenization(convert from string to List)
    lst_text = text.split()

    #remove stopwords
    if lst_stopwords is not None:
        lst_text = [word for word in lst_text if word not in
                    lst_stopwords]

     #stemming
    if flg_stemm == True:
        ps = nltk.stem.porter.PorterStemmer()
        lst_text = [ps.stem(word) for word in lst_text]

    #Lemmentization
    if flg_lemm == True:
```

```
        lem = nltk.stem.wordnet.WordNetLemmatizer()
        lst_text = [lem.lemmatize(word) for word in lst_text]

    # back to string from list
    text = " ".join(lst_text)
    return text

df['clean_text'] = df['Text'].apply(lambda x: utils_preprocess_text(x,␣
 ↪flg_stemm = False, flg_lemm=True))
```

1.5 A Glimpse into the cleansed data!

[13]:
```
df
```

[13]:
```
                                                    Text  \
0     L. Laxmikanta v State by Superintendent of Pol…
1     Homi Rajvansh v State of Maharashtra and other…
2     Direct Recruit Class Ii Engineering OfficersAs…
3     Rajinder Kumar Kindra v Delhi Administration T…
4     Kalyan and Others v State of Uttar Pradesh Sup…
…                                                   …
2909  Haryana State Cooperative Labour and others v …
2910  State of Karnataka v Chikkahottappa Alias Vara…
2911  Kilari Malakondiaah @ Malayadri and Others v S…
2912  Kanthimathy Plantations Pvt- Limited v State O…
2913  Union of India and Others v K. P. Prabhakaran …

                                              clean_text
0     l laxmikanta v state by superintendent of poli…
1     homi rajvansh v state of maharashtra and other…
2     direct recruit class ii engineering officersas…
3     rajinder kumar kindra v delhi administration t…
4     kalyan and others v state of uttar pradesh sup…
…                                                   …
2909  haryana state cooperative labour and others v …
2910  state of karnataka v chikkahottappa alias vara…
2911  kilari malakondiaah malayadri and others v sta…
2912  kanthimathy plantation pvt limited v state of …
2913  union of india and others v k p prabhakaran su…

[2914 rows x 2 columns]
```

[14]:
```
!pip install texthero==1.0.5 #texthero is a powerful nlp tool
```

```
Requirement already satisfied: texthero==1.0.5 in /opt/conda/lib/python3.7/site-
packages (1.0.5)
```

```
Requirement already satisfied: plotly-express in /opt/conda/lib/python3.7/site-
packages (from texthero==1.0.5) (0.4.1)
Requirement already satisfied: nltk in /opt/conda/lib/python3.7/site-packages
(from texthero==1.0.5) (3.2.4)
Requirement already satisfied: scikit-learn in /opt/conda/lib/python3.7/site-
packages (from texthero==1.0.5) (0.24.1)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages
(from nltk->texthero==1.0.5) (1.15.0)
Requirement already satisfied: plotly>=4.1.0 in /opt/conda/lib/python3.7/site-
packages (from plotly-express->texthero==1.0.5) (4.14.3)
Requirement already satisfied: patsy>=0.5 in /opt/conda/lib/python3.7/site-
packages (from plotly-express->texthero==1.0.5) (0.5.1)
Requirement already satisfied: scipy>=0.18 in /opt/conda/lib/python3.7/site-
packages (from plotly-express->texthero==1.0.5) (1.5.4)
Requirement already satisfied: pandas>=0.20.0 in /opt/conda/lib/python3.7/site-
packages (from plotly-express->texthero==1.0.5) (1.1.5)
Requirement already satisfied: numpy>=1.11 in /opt/conda/lib/python3.7/site-
packages (from plotly-express->texthero==1.0.5) (1.19.5)
Requirement already satisfied: statsmodels>=0.9.0 in
/opt/conda/lib/python3.7/site-packages (from plotly-express->texthero==1.0.5)
(0.12.2)
Requirement already satisfied: python-dateutil>=2.7.3 in
/opt/conda/lib/python3.7/site-packages (from pandas>=0.20.0->plotly-
express->texthero==1.0.5) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in /opt/conda/lib/python3.7/site-
packages (from pandas>=0.20.0->plotly-express->texthero==1.0.5) (2021.1)
Requirement already satisfied: retrying>=1.3.3 in /opt/conda/lib/python3.7/site-
packages (from plotly>=4.1.0->plotly-express->texthero==1.0.5) (1.3.3)
Requirement already satisfied: joblib>=0.11 in /opt/conda/lib/python3.7/site-
packages (from scikit-learn->texthero==1.0.5) (1.0.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/opt/conda/lib/python3.7/site-packages (from scikit-learn->texthero==1.0.5)
(2.1.0)
```

[15]:
```
import texthero as hero
```

1.6 TF-IDF: Term frequency–inverse document frequency

1.6.1 A numerical statistic that is intended to reflect how important a word is to a document in a corpus.

[16]:
```
df['tfidf'] = hero.do_tfidf(df['clean_text'])
```

[17]:
```
df['tfidf']
```

[17]:
```
0       [0.036527890324617486, 0.022072598945900933, 0…
1       [0.0680975915495247, 0.014402194732130617, 0.0…
2       [0.0, 0.002176426570238964, 0.0172982357520640…
3       [0.0, 0.014077931517055916, 0.0051378696136571…
4       [0.10465800676875278, 0.004918774036765286, 0…
                                 …
2909    [0.0, 0.06597640534295754, 0.01203934568864838…
2910    [0.07186571273694932, 0.016887919656505383, 0…
2911    [0.0942349275447202, 0.07402597486681295, 0.01…
2912    [0.0, 0.30147098614966705, 0.0, 0.007074728345…
2913    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0223263946041…
Name: tfidf, Length: 2914, dtype: object
```

1.6.2 Creating a test dataframe using the Query file

[18]:
```
test = pd.read_csv("/kaggle/input/legalai/Query_doc.txt",delimiter =␣
 ↪"|",header=None)
test.columns = ["AILA","NAN", "Query"]
test=test.drop(columns=["AILA","NAN"])
```

[19]:
```
test
```

[19]:
```
                                                Query
0   The appellant on February 9, 1961 was appointe…
1   The appellant before us was examined as prime …
2   This appeal arises from the judgment of the le…
3   The Petitioner was married to the Respondent N…
4   This appeal is preferred against the judgment …
5   On 19.3.1999, SI P1 along Ct. P2 went to Villa…
6   This criminal appeal is directed against the j…
7   This appeal, by special leave, has been prefer…
8   The complainant P1 filed a Special Leave Petit…
9   The four appellants, along with P1 son of P2, …
10  The detenu P1, a French national, at the relev…
11  The petitioner has been under detention pursua…
12  This is an appeal with a certificate granted b…
13  P1 is before us being aggrieved by and dissati…
14  The appellants are five in number and they hav…
15  The appellant P1 is convicted by the Additiona…
16  facts of the matter, as is evident from the pr…
17  These appeals involve a pure question of law a…
18  This appeal is preferred by the appellants aga…
19  This appeal by special leave is directed again…
20  Challenge in this appeal is to the judgment of…
21  Assailing the legal acceptability of the judgm…
22  The petitioner is a firm carrying on business …
```

```
23    These appeals are directed against the judgmen…
24    These appeals involving common questions of la…
25    The hearing before us now relates to certain o…
26    Appellant before us was detained. He is the Ma…
27    Challenge in this appeal is to the judgment of…
28    This appeal has been preferred against the jud…
29    That the deceased P1 got married to P2, the 2n…
30    This appeal by special leave is directed again…
31    On 9th May, 2004, the marriage of the daughter…
32    This is an appeal by special leave from the ju…
33    These writ petitions are filed as Public Inter…
34    Two appellants, who are brothers, along with t…
35    Interpretation and/or application of Medical B…
36    Appellants call in question legality of the ju…
37    The appellant herein is a Senior Manager in a …
38    Challenge in this appeal is to the order of a …
39    Having been selected by the Public Service Com…
40    Appellant calls in question legality of the ju…
41    This appeal arises out of the judgment dated 2…
42    Transfer Petition have been filed to transfer …
43    This petition is by the State directed against…
44    The appellants were tried for offences on the …
45    In this appeal by special leave the sole appel…
46    Challenge in this appeal is to the judgment of…
47    Whether sanction is required to initiate crimi…
48    Appellant was a Patwari working at village V1 …
49    A peculiar feature of this appeal by special l…
```

1.7 Cleanse the test data

1.7.1 We use the same methods as above to cleanse the test data

[20]:
```
test['clean_text'] = test['Query'].apply(lambda x: utils_preprocess_text(x,␣
↪flg_stemm = False, flg_lemm=True))
```

[21]:
```
test['tfidf'] = hero.do_tfidf(test['clean_text'])
```

1.8 A glimpse of how the test dataframe looks like

1.8.1 It consists of 3 columns- The original query, the cleaned text and the vectorized data

[22]:
```
test
```

```
[22]:                                             Query  \
0   The appellant on February 9, 1961 was appointe…
1   The appellant before us was examined as prime …
2   This appeal arises from the judgment of the le…
3   The Petitioner was married to the Respondent N…
4   This appeal is preferred against the judgment …
5   On 19.3.1999, SI P1 along Ct. P2 went to Villa…
6   This criminal appeal is directed against the j…
7   This appeal, by special leave, has been prefer…
8   The complainant P1 filed a Special Leave Petit…
9   The four appellants, along with P1 son of P2, …
10  The detenu P1, a French national, at the relev…
11  The petitioner has been under detention pursua…
12  This is an appeal with a certificate granted b…
13  P1 is before us being aggrieved by and dissati…
14  The appellants are five in number and they hav…
15  The appellant P1 is convicted by the Additiona…
16  facts of the matter, as is evident from the pr…
17  These appeals involve a pure question of law a…
18  This appeal is preferred by the appellants aga…
19  This appeal by special leave is directed again…
20  Challenge in this appeal is to the judgment of…
21  Assailing the legal acceptability of the judgm…
22  The petitioner is a firm carrying on business …
23  These appeals are directed against the judgmen…
24  These appeals involving common questions of la…
25  The hearing before us now relates to certain o…
26  Appellant before us was detained. He is the Ma…
27  Challenge in this appeal is to the judgment of…
28  This appeal has been preferred against the jud…
29  That the deceased P1 got married to P2, the 2n…
30  This appeal by special leave is directed again…
31  On 9th May, 2004, the marriage of the daughter…
32  This is an appeal by special leave from the ju…
33  These writ petitions are filed as Public Inter…
34  Two appellants, who are brothers, along with t…
35  Interpretation and/or application of Medical B…
36  Appellants call in question legality of the ju…
37  The appellant herein is a Senior Manager in a …
38  Challenge in this appeal is to the order of a …
39  Having been selected by the Public Service Com…
40  Appellant calls in question legality of the ju…
41  This appeal arises out of the judgment dated 2…
42  Transfer Petition have been filed to transfer …
43  This petition is by the State directed against…
44  The appellants were tried for offences on the …
45  In this appeal by special leave the sole appel…
```

```
46  Challenge in this appeal is to the judgment of…
47  Whether sanction is required to initiate crimi…
48  Appellant was a Patwari working at village V1 …
49  A peculiar feature of this appeal by special l…

                                           clean_text  \
0   the appellant on february 9 1961 wa appointed …
1   the appellant before u wa examined a prime wit…
2   this appeal arises from the judgment of the le…
3   the petitioner wa married to the respondent no…
4   this appeal is preferred against the judgment …
5   on 1931999 si p1 along ct p2 went to village v…
6   this criminal appeal is directed against the j…
7   this appeal by special leave ha been preferred…
8   the complainant p1 filed a special leave petit…
9   the four appellant along with p1 son of p2 wer…
10  the detenu p1 a french national at the relevan…
11  the petitioner ha been under detention pursuan…
12  this is an appeal with a certificate granted b…
13  p1 is before u being aggrieved by and dissatis…
14  the appellant are five in number and they have…
15  the appellant p1 is convicted by the additiona…
16  fact of the matter a is evident from the prese…
17  these appeal involve a pure question of law a …
18  this appeal is preferred by the appellant agai…
19  this appeal by special leave is directed again…
20  challenge in this appeal is to the judgment of…
21  assailing the legal acceptability of the judgm…
22  the petitioner is a firm carrying on business …
23  these appeal are directed against the judgment…
24  these appeal involving common question of law …
25  the hearing before u now relates to certain ob…
26  appellant before u wa detained he is the manag…
27  challenge in this appeal is to the judgment of…
28  this appeal ha been preferred against the judg…
29  that the deceased p1 got married to p2 the 2nd…
30  this appeal by special leave is directed again…
31  on 9th may 2004 the marriage of the daughter o…
32  this is an appeal by special leave from the ju…
33  these writ petition are filed a public interes…
34  two appellant who are brother along with their…
35  interpretation andor application of medical be…
36  appellant call in question legality of the jud…
37  the appellant herein is a senior manager in a …
38  challenge in this appeal is to the order of a …
39  having been selected by the public service com…
40  appellant call in question legality of the jud…
```

```
41   this appeal arises out of the judgment dated 2…
42   transfer petition have been filed to transfer …
43   this petition is by the state directed against…
44   the appellant were tried for offence on the al…
45   in this appeal by special leave the sole appel…
46   challenge in this appeal is to the judgment of…
47   whether sanction is required to initiate crimi…
48   appellant wa a patwari working at village v1 i…
49   a peculiar feature of this appeal by special l…

                                                tfidf
0    [0.0, 0.0, 0.0, 0.023265269274371628, 0.0, 0.0…
1    [0.03603730006560546, 0.0, 0.01801865003280273…
2    [0.0, 0.17116461074366593, 0.0, 0.108547582725…
3    [0.0, 0.09919502078555022, 0.0, 0.104844294602…
4    [0.034799206112638245, 0.12179722139423385, 0…
5    [0.07263643606978144, 0.12106072678296907, 0.0…
6    [0.0, 0.0, 0.0, 0.030927126783117872, 0.041908…
7    [0.05344425372445778, 0.12470325869040148, 0.0…
8    [0.026273195373738057, 0.19704896530303542, 0…
9    [0.05659528663050006, 0.08489292994575008, 0.0…
10   [0.012877629370496111, 0.0, 0.0128776293704961…
11   [0.057408958429978375, 0.0, 0.1148179168599567…
12   [0.0, 0.0, 0.0, 0.01654425465893878, 0.0, 0.0,…
13   [0.1279251903453413, 0.4605306852432287, 0.0, …
14   [0.0, 0.10380230512280957, 0.02076046102456191…
15   [0.020455964904773322, 0.020455964904773322, 0…
16   [0.019740354988249782, 0.0, 0.0592210649647493…
17   [0.0, 0.0, 0.0, 0.026044138243281206, 0.0, 0.0…
18   [0.07585285499889383, 0.10619399699845136, 0.0…
19   [0.04558531215274689, 0.022792656076373444, 0…
20   [0.0, 0.0, 0.0, 0.0, 0.02478303501904331, 0.03…
21   [0.0, 0.0821331014672236, 0.04693320083841348,…
22   [0.0, 0.0, 0.029320606332457776, 0.0, 0.016797…
23   [0.03355313975357944, 0.03355313975357944, 0.0…
24   [0.0, 0.0, 0.022130535231622558, 0.0, 0.0, 0.0…
25   [0.0, 0.0, 0.018166775160161464, 0.03072223120…
26   [0.0, 0.0, 0.026167802668390965, 0.02212647199…
27   [0.08738713519606371, 0.043693567598031856, 0…
28   [0.013588703081613454, 0.0, 0.0135887030816134…
29   [0.04412224836244157, 0.04412224836244157, 0.0…
30   [0.0, 0.0, 0.035122408451540986, 0.02969813694…
31   [0.038138029454161775, 0.07627605890832355, 0…
32   [0.0, 0.0, 0.0, 0.049178567007446584, 0.022213…
33   [0.0, 0.036177706723484984, 0.0, 0.05098409324…
34   [0.04868913246463977, 0.036516849348479825, 0…
35   [0.01687138723918626, 0.0, 0.0, 0.0, 0.0, 0.04…
```

```
36    [0.08000542904710295, 0.14000950083243016, 0.0…
37    [0.0, 0.02853336587761116, 0.0, 0.048253399727…
38    [0.015486704304197043, 0.1858404516503645, 0.0…
39    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.08532376…
40    [0.045849443460218915, 0.1681146260208027, 0.0…
41    [0.04194241570193545, 0.0, 0.0, 0.141859474909…
42    [0.0, 0.02252064263190128, 0.02252064263190128…
43    [0.0, 0.023058118744510716, 0.0, 0.05849113414…
44    [0.05883782866075069, 0.05883782866075069, 0.0…
45    [0.018994276160114676, 0.0759771046404587, 0.0…
46    [0.014804194628789225, 0.1628461409166815, 0.0…
47    [0.0, 0.07220487883970768, 0.03610243941985384…
48    [0.04185436650088952, 0.0, 0.0, 0.070780835523…
49    [0.03355805746669458, 0.0, 0.01677902873334729…
```

2 Vectorizing the data using Tfidf Vectorizer

[23]:
```
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import TfidfVectorizer

# Vectorise the data
vec = TfidfVectorizer()

X = vec.fit_transform([df['clean_text'][0]]) # `X` will now be a TF-IDF␣
↪representation of the data, the first row of `X` corresponds to the first␣
↪sentence in `data`
Y = vec.transform([test['clean_text'][0]]) # 'Y' will now be a TF-IDF␣
↪representation of the data
```

2.1 Finding Cosine Similarity for each of the Queries to each of the training samples

2.1.1 For each of the 50 queries, we iterate over all the prior cases and find the one which has the highest cosine similarity with the that query. We then append both these values into separate arrays.

[24]:
```
max_index_array = []
max_similarity_score_array = []

for i in range(len(test)):
    Y = vec.fit_transform([test['clean_text'][i]])

    max_similarity = 0
```

```
    max_index = -1
    for j in range(len(df)):

        X = vec.transform([df['clean_text'][j]])
        S = cosine_similarity(X,Y)
        #print(S[i][0])
        if (S[0][0]>max_similarity):
            max_similarity = S[0][0]
            max_index = j

    max_index_array.append(max_index)
    max_similarity_score_array.append(max_similarity)
```

[28]:
```
print(max_index_array) # the index of the prior document with the highest␣
↪cosine similarity for each of the 50 queries
```

```
[232, 822, 339, 2694, 663, 1089, 1428, 1559, 668, 906, 2770, 2379, 2525, 1238,
1138, 2490, 2604, 2187, 2774, 2394, 1940, 402, 2110, 1596, 1244, 822, 295, 28,
1141, 1598, 918, 1534, 2285, 1688, 1858, 746, 2507, 1179, 2180, 2315, 2575,
2694, 2390, 2073, 2649, 1381, 2575, 1873, 79, 822]
```

[29]:
```
print(max_similarity_score_array) # the value of the cosine similarity for each␣
↪of the 50 queries
```

```
[0.9529367001327095, 0.9409197374568615, 0.9370746787515178, 0.9434257480309546,
0.9284010961694416, 0.9178189599766118, 0.9495154131757083, 0.940872962834488,
0.941410860601876, 0.8997674205562887, 0.9552130351589725, 0.8013936362070003,
0.9263123745716466, 0.9002578877916514, 0.978782413885863, 0.8825793489183309,
0.9498736471895454, 0.9468904718093757, 0.96733592138956, 0.949479776054762,
0.9472911853507456, 0.9709765904549609, 0.9347725228457751, 0.9506069528132177,
0.9326516683507472, 0.9613735520602388, 0.893275688670986, 0.9114074610301138,
0.9482734197666621, 0.9563297685390909, 0.9536303583580877, 0.9131507471776771,
0.9488887516914465, 0.9399663087901251, 0.9493459348775186, 0.9017758603772148,
0.9448850183294216, 0.9361226750802891, 0.9567124763482466, 0.9129556972813528,
0.9592978985338997, 0.9136064028313299, 0.8767267584477333, 0.9379712924011954,
0.945967620620473, 0.922772599949533, 0.964234556401232, 0.9355647739786285,
0.880158553263502, 0.9387155582204889]
```