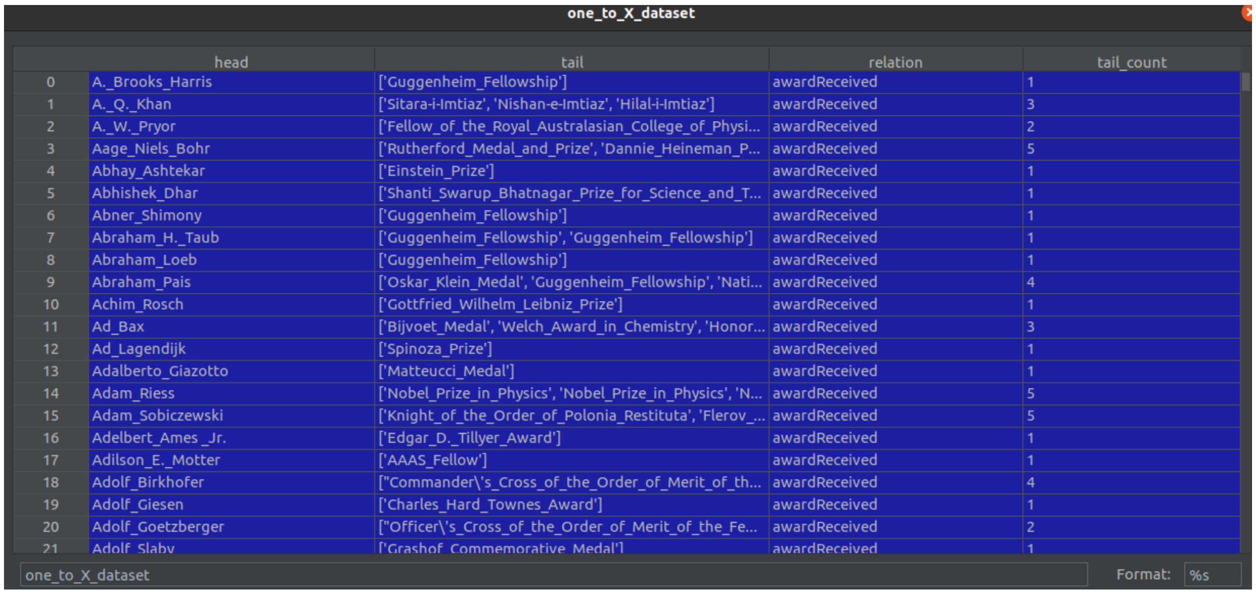
For Dbpedia dataset (for train dataset)

One to one = 13530
One to many = 3810
Many to one = 3972

For Wiki dataset (Train dataset)
One to many = 6138
Many to one = 1930
One to one = 6547



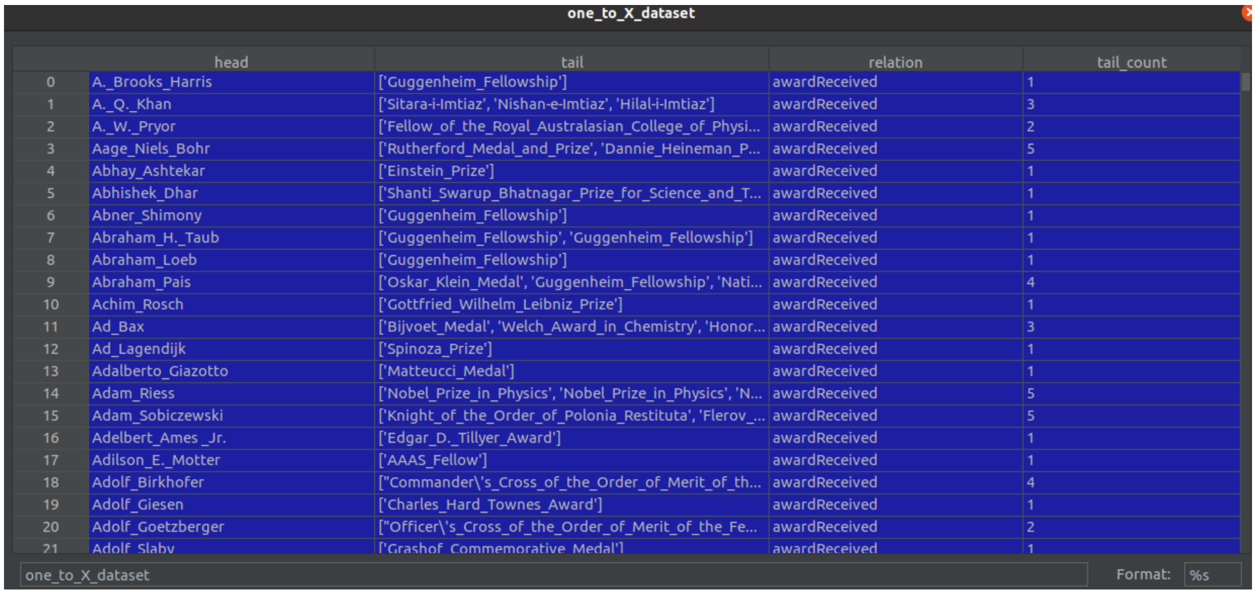
one_to_X_dataset

	head	tail	relation	tail_count
0	A._Brooks_Harris	['Guggenheim_Fellowship']	awardReceived	1
1	A._Q._Khan	['Sitara-i-Imtiaz', 'Nishan-e-Imtiaz', 'Hilal-i-Imtiaz']	awardReceived	3
2	A._W._Pryor	['Fellow_of_the_Royal_Australasian_College_of_Physi...	awardReceived	2
3	Aage_Niels_Bohr	['Rutherford_Medal_and_Prize', 'Dannie_Heineman_P...	awardReceived	5
4	Abhay_Ashtekar	['Einstein_Prize']	awardReceived	1
5	Abhishek_Dhar	['Shanti_Swarup_Bhatnagar_Prize_for_Science_and_T...	awardReceived	1
6	Abner_Shimony	['Guggenheim_Fellowship']	awardReceived	1
7	Abraham_H._Taub	['Guggenheim_Fellowship', 'Guggenheim_Fellowship']	awardReceived	1
8	Abraham_Loeb	['Guggenheim_Fellowship']	awardReceived	1
9	Abraham_Pais	['Oskar_Klein_Medal', 'Guggenheim_Fellowship', 'Nati...	awardReceived	4
10	Achim_Rosch	['Gottfried_Wilhelm_Leibniz_Prize']	awardReceived	1
11	Ad_Bax	['Bijvoet_Medal', 'Welch_Award_in_Chemistry', 'Honor...	awardReceived	3
12	Ad_Lagendijk	['Spinoza_Prize']	awardReceived	1
13	Adalberto_Giazotto	['Matteucci_Medal']	awardReceived	1
14	Adam_Riess	['Nobel_Prize_in_Physics', 'Nobel_Prize_in_Physics', 'N...	awardReceived	5
15	Adam_Sobiczewski	['Knight_of_the_Order_of_Polonia_Restituta', 'Flerov_...	awardReceived	5
16	Adelbert_Ames _Jr.	['Edgar_D._Tillyer_Award']	awardReceived	1
17	Adilson_E._Motter	['AAAS_Fellow']	awardReceived	1
18	Adolf_Birkhofer	["Commander\'s_Cross_of_the_Order_of_Merit_of_th...	awardReceived	4
19	Adolf_Giesen	['Charles_Hard_Townes_Award']	awardReceived	1
20	Adolf_Goetzberger	["Officer\'s_Cross_of_the_Order_of_Merit_of_the_Fe...	awardReceived	2
21	Adolf_Slaby	['Grashof_Commemorative_Medal']	awardReceived	1

one_to_X_dataset Format: %s

One to oneimport numpy as np

```
import  pandas as pd

data = 
pd.read_table('/home/tansen/my_files/thesisUpdatedNew/dataset/dbpediadata/result/train_origi
nal.txt' , header=None)
data = data[[0,1,2]]
data.columns = ['head','relation','tail']
unique_relations = data['relation'].unique()

one_to_X_dataset = pd.DataFrame()
X_to_one_dataset = pd.DataFrame()
```

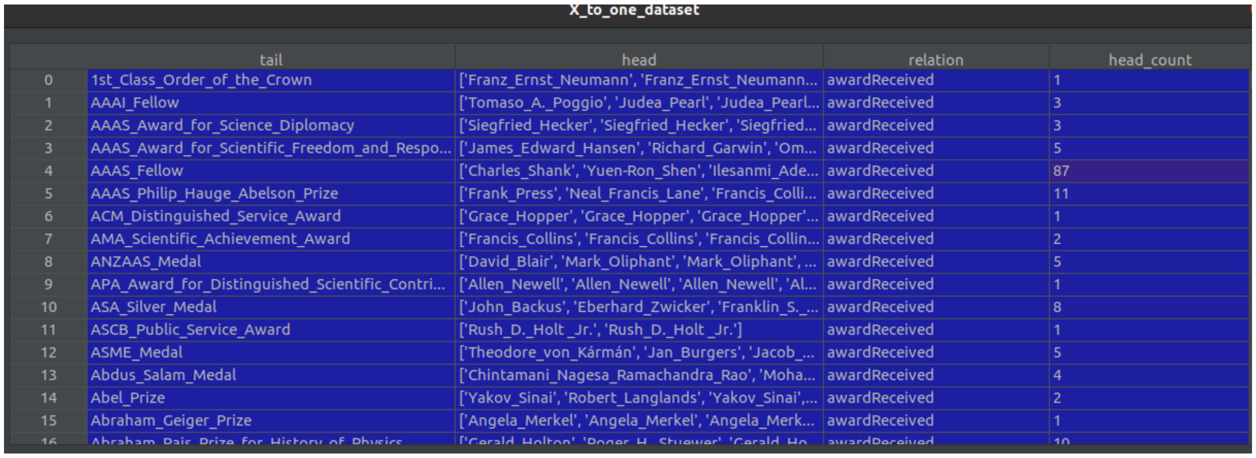


one_to_X_dataset

	head	tail	relation	tail_count
0	A._Brooks_Harris	['Guggenheim_Fellowship']	awardReceived	1
1	A._Q._Khan	['Sitara-i-Imtiaz', 'Nishan-e-Imtiaz', 'Hilal-i-Imtiaz']	awardReceived	3
2	A._W._Pryor	['Fellow_of_the_Royal_Australasian_College_of_Physi...	awardReceived	2
3	Aage_Niels_Bohr	['Rutherford_Medal_and_Prize', 'Dannie_Heineman_P...	awardReceived	5
4	Abhay_Ashtekar	['Einstein_Prize']	awardReceived	1
5	Abhishek_Dhar	['Shanti_Swarup_Bhatnagar_Prize_for_Science_and_T...	awardReceived	1
6	Abner_Shimony	['Guggenheim_Fellowship']	awardReceived	1
7	Abraham_H._Taub	['Guggenheim_Fellowship', 'Guggenheim_Fellowship']	awardReceived	1
8	Abraham_Loeb	['Guggenheim_Fellowship']	awardReceived	1
9	Abraham_Pais	['Oskar_Klein_Medal', 'Guggenheim_Fellowship', 'Nati...	awardReceived	4
10	Achim_Rosch	['Gottfried_Wilhelm_Leibniz_Prize']	awardReceived	1
11	Ad_Bax	['Bijvoet_Medal', 'Welch_Award_in_Chemistry', 'Honor...	awardReceived	3
12	Ad_Lagendijk	['Spinoza_Prize']	awardReceived	1
13	Adalberto_Giazotto	['Matteucci_Medal']	awardReceived	1
14	Adam_Riess	['Nobel_Prize_in_Physics', 'Nobel_Prize_in_Physics', 'N...	awardReceived	5
15	Adam_Sobiczewski	['Knight_of_the_Order_of_Polonia_Restituta', 'Flerov_...	awardReceived	5
16	Adelbert_Ames_Jr.	['Edgar_D._Tillyer_Award']	awardReceived	1
17	Adilson_E._Motter	['AAAS_Fellow']	awardReceived	1
18	Adolf_Birkhofer	["Commander\'s_Cross_of_the_Order_of_Merit_of_th...	awardReceived	4
19	Adolf_Giesen	['Charles_Hard_Townes_Award']	awardReceived	1
20	Adolf_Goetzberger	["Officer\'s_Cross_of_the_Order_of_Merit_of_the_Fe...	awardReceived	2
21	Adolf_Slaby	['Grashof_Commemorative_Medal']	awardReceived	1

one_to_X_dataset Format: %s

One to many

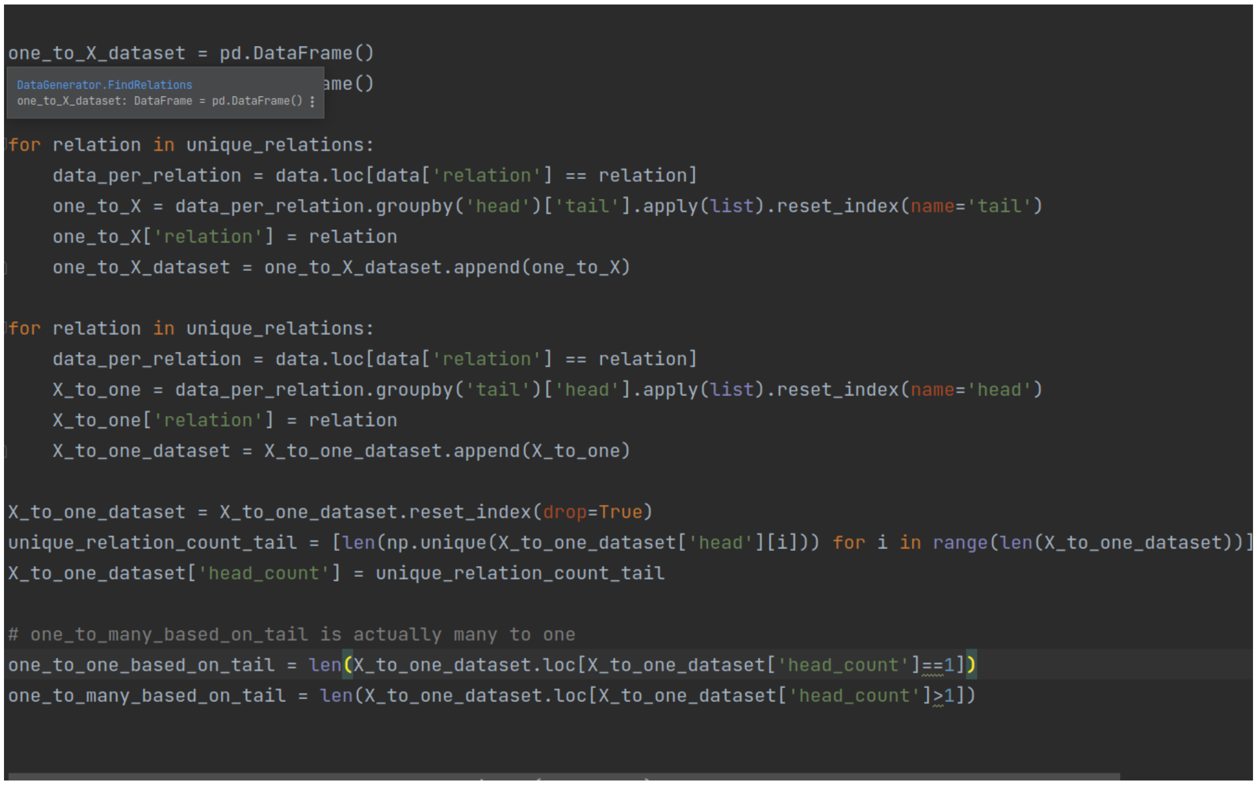


X_to_one_dataset

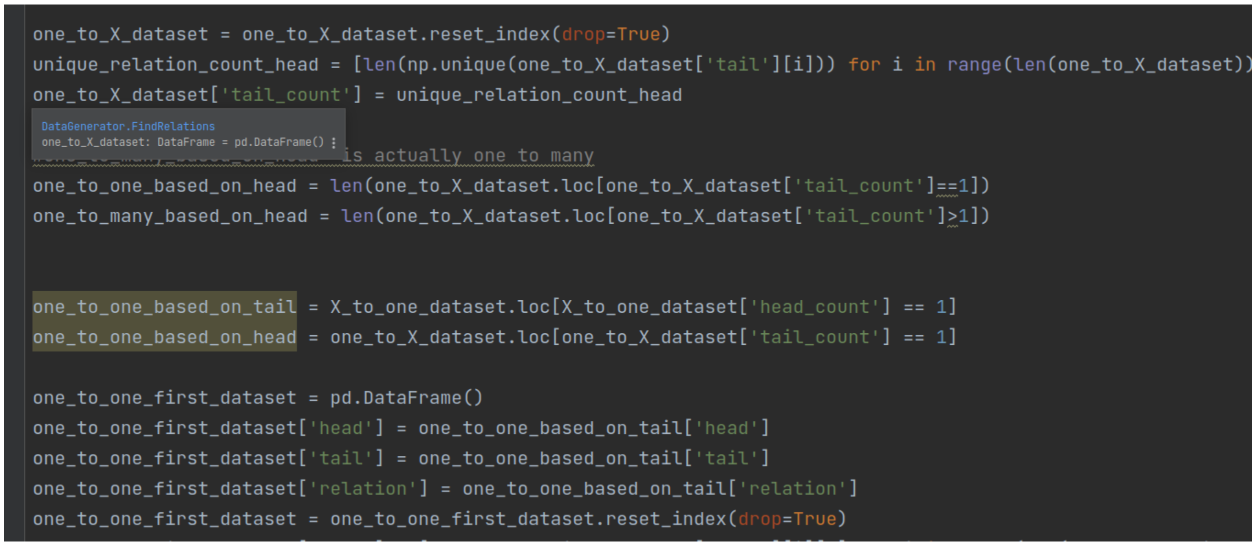
	tail	head	relation	head_count
0	1st_Class_Order_of_the_Crown	['Franz_Ernst_Neumann', 'Franz_Ernst_Neumann...	awardReceived	1
1	AAAI_Fellow	['Tomaso_A._Poggio', 'Judea_Pearl', 'Judea_Pearl...	awardReceived	3
2	AAAS_Award_for_Science_Diplomacy	['Siegfried_Hecker', 'Siegfried_Hecker', 'Siegfried...	awardReceived	3
3	AAAS_Award_for_Scientific_Freedom_and_Respo...	['James_Edward_Hansen', 'Richard_Garwin', 'Om...	awardReceived	5
4	AAAS_Fellow	['Charles_Shank', 'Yuen-Ron_Shen', 'Ilesanmi_Ade...	awardReceived	87
5	AAAS_Philip_Hauge_Abelson_Prize	['Frank_Press', 'Neal_Francis_Lane', 'Francis_Colli...	awardReceived	11
6	ACM_Distinguished_Service_Award	['Grace_Hopper', 'Grace_Hopper', 'Grace_Hopper'...	awardReceived	1
7	AMA_Scientific_Achievement_Award	['Francis_Collins', 'Francis_Collins', 'Francis_Collin...	awardReceived	2
8	ANZAAS_Medal	['David_Blair', 'Mark_Oliphant', 'Mark_Oliphant', ...	awardReceived	5
9	APA_Award_for_Distinguished_Scientific_Contri...	['Allen_Newell', 'Allen_Newell', 'Allen_Newell', 'Al...	awardReceived	1
10	ASA_Silver_Medal	['John_Backus', 'Eberhard_Zwicker', 'Franklin_S._...	awardReceived	8
11	ASCB_Public_Service_Award	['Rush_D._Holt_Jr.', 'Rush_D._Holt_Jr.']	awardReceived	1
12	ASME_Medal	['Theodore_von_Kármán', 'Jan_Burgers', 'Jacob_...	awardReceived	5
13	Abdus_Salam_Medal	['Chintamani_Nagesa_Ramachandra_Rao', 'Moha...	awardReceived	4
14	Abel_Prize	['Yakov_Sinai', 'Robert_Langlands', 'Yakov_Sinai',...	awardReceived	2
15	Abraham_Geiger_Prize	['Angela_Merkel', 'Angela_Merkel', 'Angela_Merk...	awardReceived	1
16	Abraham_Pais_Prize_for_History_of_Physics	['Gerald_Holton', 'Roger_H._Stuewer', 'Gerald_Ho...	awardReceived	10

Many to one

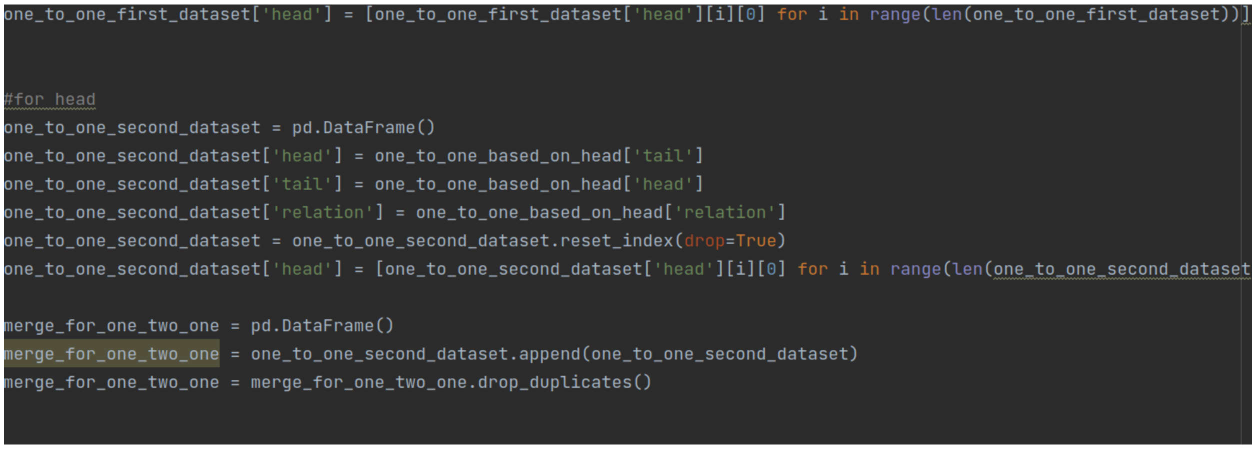
Code



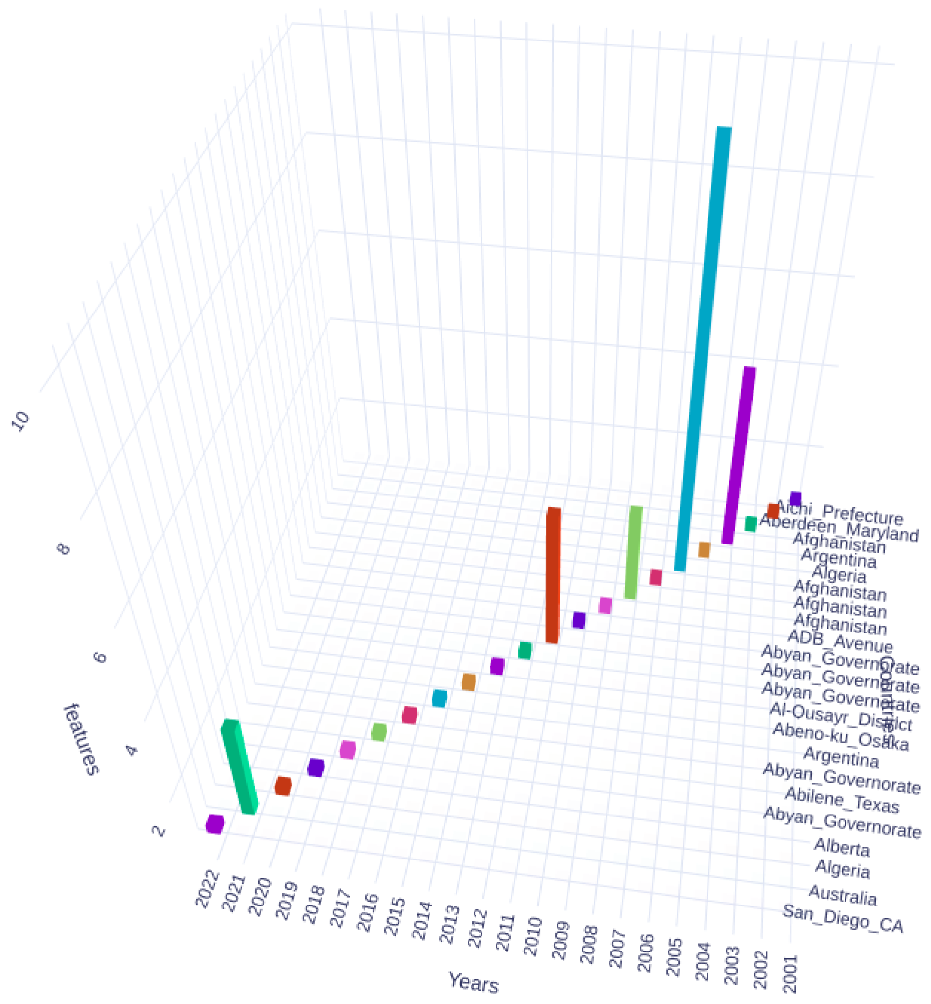
```
one_to_X_dataset = pd.DataFrame()
DataGenerator.FindRelations
one_to_X_dataset: DataFrame = pd.DataFrame() ⋮   ame()

for relation in unique_relations:
    data_per_relation = data.loc[data['relation'] == relation]
    one_to_X = data_per_relation.groupby('head')['tail'].apply(list).reset_index(name='tail')
    one_to_X['relation'] = relation
    one_to_X_dataset = one_to_X_dataset.append(one_to_X)

for relation in unique_relations:
    data_per_relation = data.loc[data['relation'] == relation]
    X_to_one = data_per_relation.groupby('tail')['head'].apply(list).reset_index(name='head')
    X_to_one['relation'] = relation
    X_to_one_dataset = X_to_one_dataset.append(X_to_one)

X_to_one_dataset = X_to_one_dataset.reset_index(drop=True)
unique_relation_count_tail = [len(np.unique(X_to_one_dataset['head'][i])) for i in range(len(X_to_one_dataset))]
X_to_one_dataset['head_count'] = unique_relation_count_tail

# one_to_many_based_on_tail is actually many to one
one_to_one_based_on_tail = len(X_to_one_dataset.loc[X_to_one_dataset['head_count']==1])
one_to_many_based_on_tail = len(X_to_one_dataset.loc[X_to_one_dataset['head_count']>1])
```



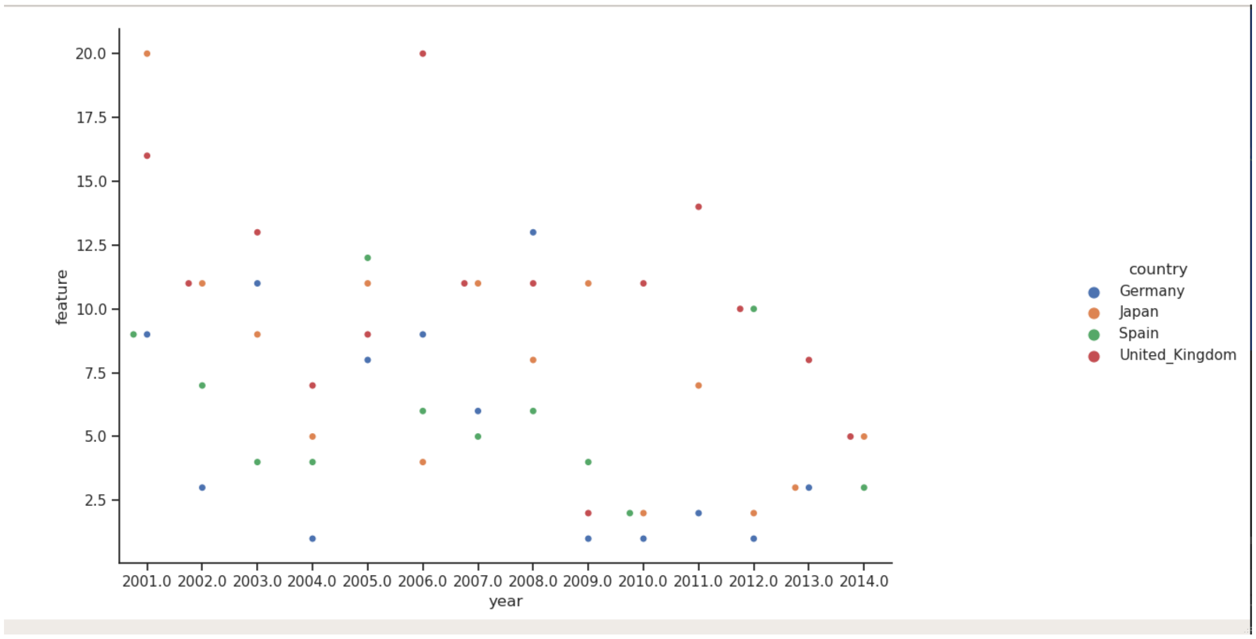
```
one_to_X_dataset = one_to_X_dataset.reset_index(drop=True)
unique_relation_count_head = [len(np.unique(one_to_X_dataset['tail'][i])) for i in range(len(one_to_X_dataset))
one_to_X_dataset['tail_count'] = unique_relation_count_head

DataGenerator.FindRelations
one_to_X_dataset: DataFrame = pd.DataFrame() ⋮   is actually one to many
one_to_one_based_on_head = len(one_to_X_dataset.loc[one_to_X_dataset['tail_count']==1])
one_to_many_based_on_head = len(one_to_X_dataset.loc[one_to_X_dataset['tail_count']>1])


one_to_one_based_on_tail = X_to_one_dataset.loc[X_to_one_dataset['head_count'] == 1]
one_to_one_based_on_head = one_to_X_dataset.loc[one_to_X_dataset['tail_count'] == 1]

one_to_one_first_dataset = pd.DataFrame()
one_to_one_first_dataset['head'] = one_to_one_based_on_tail['head']
one_to_one_first_dataset['tail'] = one_to_one_based_on_tail['tail']
one_to_one_first_dataset['relation'] = one_to_one_based_on_tail['relation']
one_to_one_first_dataset = one_to_one_first_dataset.reset_index(drop=True)
```


```
one_to_one_first_dataset['head'] = [one_to_one_first_dataset['head'][i][0] for i in range(len(one_to_one_first_dataset))]


#for head
one_to_one_second_dataset = pd.DataFrame()
one_to_one_second_dataset['head'] = one_to_one_based_on_head['tail']
one_to_one_second_dataset['tail'] = one_to_one_based_on_head['head']
one_to_one_second_dataset['relation'] = one_to_one_based_on_head['relation']
one_to_one_second_dataset = one_to_one_second_dataset.reset_index(drop=True)
one_to_one_second_dataset['head'] = [one_to_one_second_dataset['head'][i][0] for i in range(len(one_to_one_second_dataset

merge_for_one_two_one = pd.DataFrame()
merge_for_one_two_one = one_to_one_second_dataset.append(one_to_one_second_dataset)
merge_for_one_two_one = merge_for_one_two_one.drop_duplicates()
```

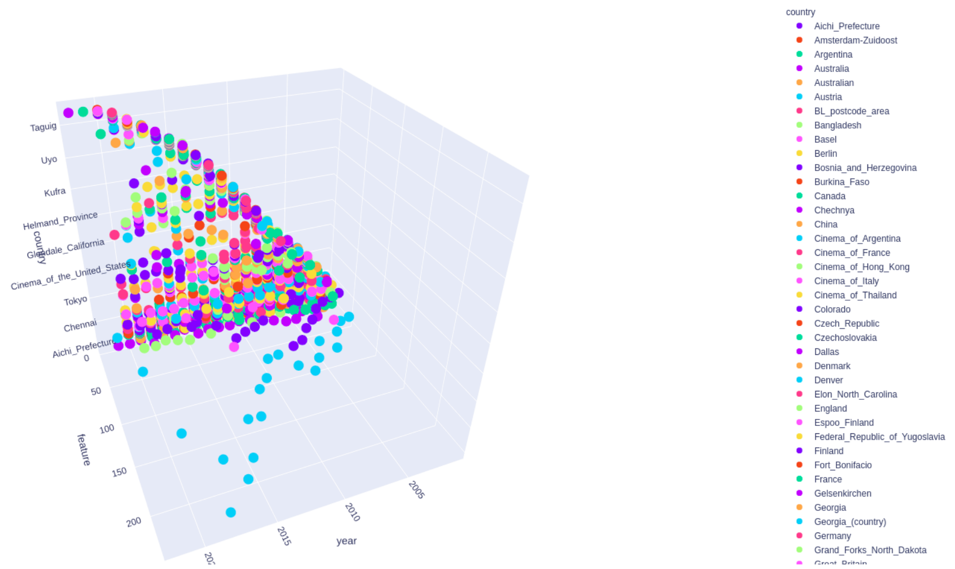
Visualization

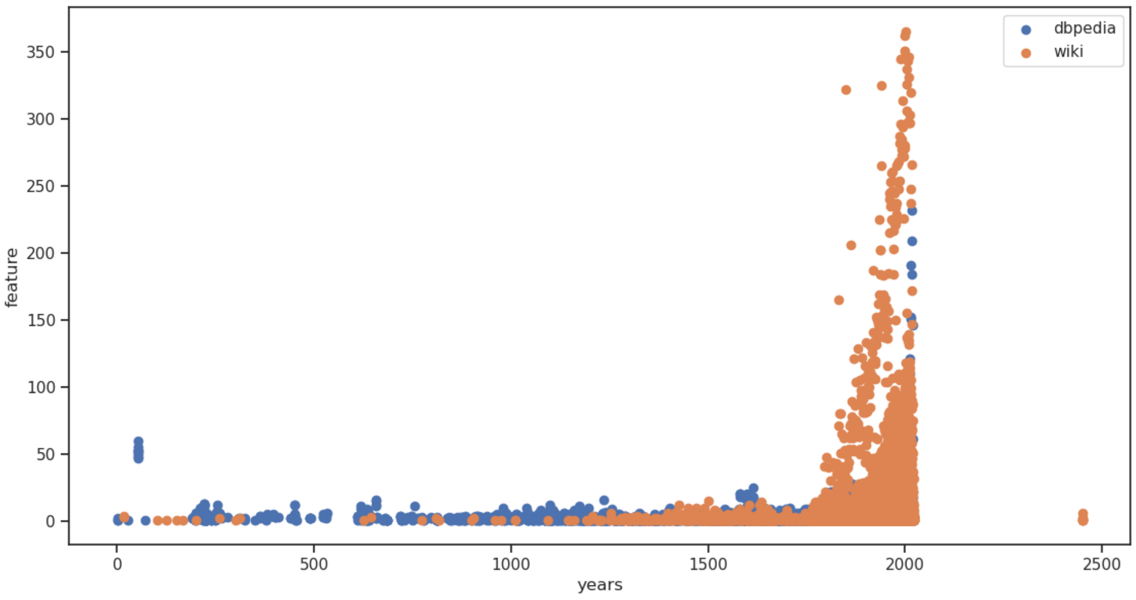


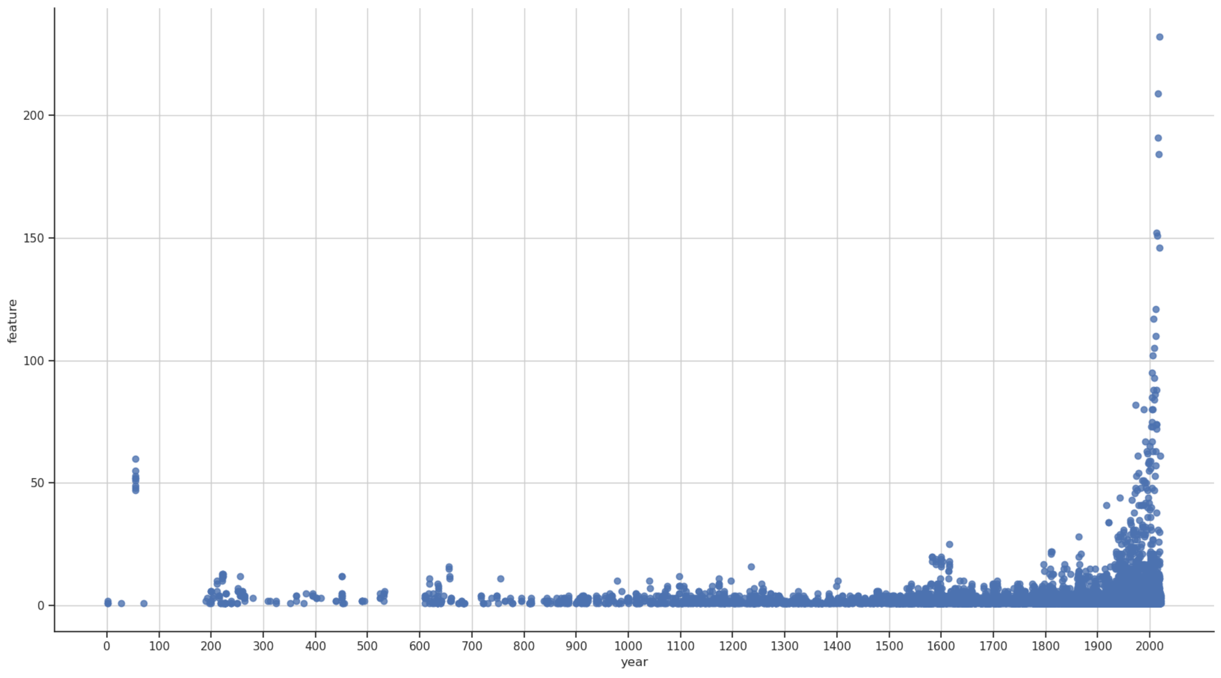
features
2
4
6
8
10
Years
2022
2021
2020
2019
2018
2017
2016
2015
2014
2013
2012
2011
2010
2009
2008
2007
2006
2005
2004
2003
2002
2001
Aichi_Prefecture
Aberdeen_Maryland
Afghanistan
Argentina
Algeria
Afghanistan
Afghanistan
Afghanistan
ADB_Avenue
Abyan_Governorate
Abyan_Governorate
Abyan_Governorate
Al-Ousayr_District
Abeno-ku_Osaka
Argentina
Abyan_Governorate
Abilene_Texas
Abyan_Governorate
Alberta
Algeria
Australia
San_Diego_CA



3D Scatter Plot




dbpedia
wiki
feature
years
0
500
1000
1500
2000
2500
0
50
100
150
200
250
300
350


feature
year
0
50
100
150
200
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000