ETL-Helper

A little helper to help find the probable title

In [1]:

```
#CORRECTION FOR TITLES
#!pip install PyYAML
#!pip install fold_to_ascii
```

In [2]:

```
import MungingOps as tt
import pandas as pd
import warnings
```

In [3]:

```
# Flexibility, rapid responce:
# This is not an oraculus, you not magicaly obtain the responce
# Manipulate the number to manipulate the result
#
#
# Flexibility, Razonable explication:
# Flexibility refers to the range of bad response can be accepted;
# this range is a quasy fuzzy response, not a binary logic
# then the Jaccard Distance say this string is most ok or not ok,
# traditionaly the jaccard distance in 0.8 is acceptable, but
# sometimes is necesary more low number to obtain a responce
#
Flexibility = 0.6
#
warnings.filterwarnings('ignore')
```

Load the Dictionary

this file contains the relation of malformed tiles and their respective right titles

In [4]:

```
RAW = pd.read_csv('DictionaryTitles.csv')
```

Drop nulls

In [5]:

```
try:
    RAW.dropna(thresh=1, inplace=True)
except: None
try:
    RAW.dropna(thresh=2, inplace=True)
except: None
```

Work Safe with copy of Data

In [6]:

```
df=RAW
```

Load the Exceptions to evaluate

In [7]:

```
EF = pd.read_csv('ExceptionTitles.csv')
ExceptionTitles = EF.copy()
```

Drop Nulls

In [8]:

```
ExceptionTitles.dropna(inplace=True)
```

Clean the text to common english (130 ascii non unicoide) characters without symbols

In [9]:

```
chng=[]
for i in range(len(ExceptionTitles['Exceptions'])):
    chng.append( tt.ExtraWhite(str(ExceptionTitles['Exceptions'][i]).replace('/','')
ExceptionTitles['ExceptionsVector'] = chng
del chng
```

In [10]:

```
chng=[]
for i in range(len(df['GP_Title'])):
    chng.append( tt.ExtraWhite(str(df['GP_Title'][i]).replace('/','').replace(':',''
df['GP_Title'] = chng
del chng
```

Calculate the Jaccard Distance by each Exception for each Title

In [11]:

```
# Calculate the Jaccard Distance by each Exception for each Title
# Select the maximum Jaccard
# if Jaccard is > Flexibility the make a Suggestion
# si no pues no -0_o-
#
Exceptions=[]
Suggested=[]
Correction=[]
Correction2=[]
#
for i in range(len(ExceptionTitles)):
    try:
        Comparison={}
        k= ExceptionTitles['ExceptionsVector'][i]
        for j in range(len(df['GP_Title'])):
            m = df['GP_Title'][j]
            Comparison[m] = (tt.JaccardDistance(k,m))
            JaccardValues = max(zip(Comparison.values(),Comparison.keys()))
            MaxJaccard = JaccardValues[1]
        if JaccardValues[0] > Flexibility:
            Exceptions.append(k)
            Suggested.append(MaxJaccard)
            Correction.append( df[ df['GP_Title'] == MaxJaccard ]['Title'].values[0]
            Correction2.append( df[ df['GP_Title'] == MaxJaccard ]['TitleID'].values
        else:
            Exceptions.append(k)
            Suggested.append(None)
            Correction.append(None)
            Correction2.append(None)
    except:
            Exceptions.append(k)
            Suggested.append(None)
            Correction.append(None)
            Correction2.append(None)
```

In [12]:

```
# Add the results to the dataframe
#
resultado= pd.DataFrame()
resultado['Exceptions'] = Exceptions
resultado['Suggested'] = Suggested
resultado['Title'] = Correction
resultado['TitleID'] = Correction2
```

Delete the unnecesary working data

In [13]:

```
# Delete the unnecesary working data
#
del df
del ExceptionTitles
```

Save to new file

In [14]:

```
# Save to new file
# Enable in prod
resultado.to_excel('ExceptionResults.xlsx')
```