Library for language detection

`By Fábio Bif Goularte` *(fabio.goularte@gmail.com)*

Import modules

In [4]:
```
import collections
import math
import os
import pandas as pd
import numpy as np
```

Detecting the language

Set the language of the document to detecting (line in[6]) with a code from the table below (column Language code).

Language	Language code	Language	Language code
Afrikaans	af	Italian	it
German	de	Japanese	ja
English	en	Korean	ko
Spanish	es	Portuguese (Brazil)	pt-BR
Hindi	hi	Chinese, Mandarin (Simplified)	zh-Hans

`Note:` It is possible to check other languages from those listed in the table above. Thus, provide a document in the desired language and run the Training Testing file to create the test document.

In [13]:
```
#Code of language tested, e.g. 'pt-Br' to Portuguese
languageDoc = 'pt-BR'

#Load the n-gram model selected to test the language document (testing folder)
df_test = pd.read_json('testing/'+languageDoc+'.json', orient='columns')
df_test.columns=['n-gramas','freq_doc']

languages = os.listdir('testing')

#Load the n-grams models used to train the classifier according to the files in testing (training folder)
for langTrain in languages:
    langTrain  = langTrain.split('.')
    df_train   = pd.read_json('training/'+langTrain[0]+'.json', orient='columns')
    df_test    = df_test.merge(df_train,how='left',on='n-gramas')
```

Shows the n-gram models based on the documents in the testing folder

In [6]:
```
df_test.head()
```

Out[6]:

	n-gramas	freq_doc	freq_af	freq_de	freq_en	freq_es	freq_hi	freq_it	freq_ja	freq_ko	freq_pt-BR	freq_zh-Hans
0	a	445	605.0	525.0	705.0	1083.0	NaN	1004.0	NaN	NaN	949.0	1.0
1	e	391	1686.0	1751.0	1078.0	1301.0	NaN	1222.0	NaN	NaN	1181.0	NaN
2	s	327	444.0	502.0	465.0	675.0	NaN	511.0	NaN	NaN	705.0	NaN
3	o	307	414.0	173.0	706.0	828.0	NaN	935.0	NaN	NaN	959.0	NaN
4	i	243	751.0	766.0	698.0	736.0	NaN	1439.0	NaN	NaN	764.0	3.0

In [7]:
```
df_new=df_test.copy()
```

TF-IDF

In [8]:
```
coll = list(df_test.columns)
coll.pop(0)

#Calculating TF-IDF per n-grams on the selected test document and on the documents from training folder
for w in coll:
    df_new[w]=df_test[w]/len(df_test)*np.log10(len(coll)/df_test.count(axis='columns'))
```

In [9]:
```
df_new.head()
```

Out[9]:

	n-gramas	freq_doc	freq_af	freq_de	freq_en	freq_es	freq_hi	freq_it	freq_ja	freq_ko	freq_pt-BR	freq_zh-Hans
0	a	0.038322	0.052101	0.045211	0.060712	0.093264	NaN	0.086461	NaN	NaN	0.081725	0.000086
1	e	0.053435	0.230413	0.239296	0.147322	0.177798	NaN	0.167002	NaN	NaN	0.161399	NaN
2	s	0.044689	0.060678	0.068605	0.063548	0.092247	NaN	0.069835	NaN	NaN	0.096347	NaN
3	o	0.041955	0.056578	0.023643	0.096484	0.113157	NaN	0.127780	NaN	NaN	0.131060	NaN
4	i	0.020926	0.064674	0.065965	0.060110	0.063382	NaN	0.123922	NaN	NaN	0.065793	0.000258

Function used to calculate the cosine similarity

In [10]:
```
#Calculating cosine similarity
def similarity(docA,docB):
    numerator = 0
    docA      = docA.replace(np.nan,0)
    docB      = docB.replace(np.nan,0)
    den_1     = math.sqrt(sum([docA[i]**2 for i in range(0,len(docA))]))
    den_2     = math.sqrt(sum([docB[i]**2 for i in range(0,len(docB))]))

    for i in range(0,len(docA)): numerator=numerator+docA[i]*docB[i]

    denumerator = den_1*den_2
    if denumerator == 0 : denumerator = 0.0001

    return numerator/denumerator
```

In [11]:
```
coll.pop(0)
classifier = {}

#Calculating the cosine similarity between 'languageDoc' and the other documents
for w in coll:
    classifier[w] = similarity(df_new['freq_doc'],df_new[w])
```

Shown the results

In [12]:
```
print('Doc tested: '+languageDoc)
print('\nClassification by similarity:')

for key, value in sorted(classifier.items(), key=lambda item: item[1], reverse=True):
    print("%s: %s" % (key, value))
```

```
Doc tested: pt-BR

Classification by similarity:
freq_pt-BR: 0.9512008768057516
freq_es: 0.9180541189217364
freq_it: 0.8732348939396601
freq_en: 0.8575734253085008
freq_de: 0.7841373356480694
freq_af: 0.7555665053842838
freq_zh-Hans: 0.21549506976710367
freq_hi: 0.0
freq_ja: 0.0
freq_ko: 0.0
```

In [ ]:
```

```