**This version of ner.py does not utilize the English Names and English-Named-Locations. Utilizing English names provides more data for us, but it is not accurate since there are nearly 190k unchecked entries.**

**Datasets**

You can see the raw versions of the data collected in the folder “Raw Data”. Here there are 3 subfolders:

1. Locations Folder
2. Names Folder
3. Organizations Folder
4. Locations Folder:

There are 2 subfolders also.

1. Turkey-Wide

🡪 Data is retrieved from:

<https://github.com/life/il-ilce-mahalle-sokak-cadde-sql>

<https://www.gencayyildiz.com/blog/ms-sql-server-ulke-sehir-ilce-semt-ve-mahalle-veritabani/>

🡪 Data is read and formatted by the python file in the folder.

🡪 Resulted text files are named as =

“location\_data\_il.txt”

“location\_data\_ilçe.txt”

1. World-Wide

🡪 Data is retrieved from: <https://simplemaps.com/data/world-cities>

🡪 Data is read and formatted by the python file in the folder.

🡪 Resulted text file is named as = “location\_data\_world.txt”

1. Google Translate.py

There is also a python file named “Google Translate” which takes the English World-Wide data and translates it into Turkish, since we are using Turkish in our ner system. This python code uses Google Translate API for python and takes a bit long time to iterate over all the data.

🡪 Resulted text file is named as = “location\_data\_world\_turkish.txt”

1. Names Folder:
2. Turkish Names:

🡪 Data is retrieved from: <https://gist.github.com/ismailbaskin/1325813>

🡪 Data is read and formatted by the python file in the folder.

🡪 Resulted text file is named as = “names\_data\_turkish.txt”

1. English Names:

🡪 Data is retrieved from: <https://data.world/len/us-first-names-database>

🡪 Data is read and formatted by the python file in the folder.

🡪 Resulted text file is named as = “names\_data\_english.txt”

1. Organizations Folder:
2. Banks in Turkey:  
   🡪 Data is retrieved from:

<https://ipfs.io/ipfs/QmR1gzPYUwxEUWHbeRggZzfYy5Fxsd8Qc7hXUUnJQwxrZq/wiki/Türkiye%27deki_bankalar_listesi.html>

🡪 Data is read and formatted by the python file in the folder.

🡪 Resulted text file is named as = “organization\_turkish\_banks.txt”

1. Known Organizations:
2. Top Organizations World-Wide:

🡪 Data is retrieved from: <https://www.forbes.com/global2000/#3a6123fb335d>

🡪 Data is read and formatted by the python file in the folder.

🡪 Resulted text file is named as = “organization\_top\_companies.txt”

1. Top Organizations Turkey-Wide:

🡪 Data is retrieved from: <https://www.fortuneturkey.com/fortune500>

🡪 Data is read and formatted by the python file in the folder.

🡪 Resulted text file is named as = “organization\_turkey\_top.txt”

1. Turkish Government Organizations

🡪 Retrieved from: <https://www.ab.gov.tr/_2926.html>

🡪 Resulted text is named as = “organization\_turkish\_kurum.txt”

**In the end all databases are in the folder called Databases!!!**

**Regular Expressions**

1. Names:

RE for searching names:

[A-ZÇĞİÖŞÜ][a-zçğıöşü]\*\s+[A-ZÇĞİÖŞÜ][A-ZÇĞİÖŞÜa-zçğıöşü]\*(?:\s+[A-ZÇĞİÖŞÜ][A-ZÇĞİÖŞÜa-zçğıöşü]\*){1,4}

* This regex is searching for the person names.
  + Name Surname
  + Name (Middle-Name)\*1 Surname
  + Name (Middle-Name)\*2 Surname
  + Name (Middle-Name)\*3 Surname
  + Name (Middle-Name)\*4 Surname

[A-ZÇĞİÖŞÜ]\w+

* A basic name search for just Name.

(?<='+ unvan + r')\w\*\s+[A-ZÇĞİÖŞÜ][a-zçğıöşü]\*\s+[A-ZÇĞİÖŞÜ][a-zçğıöşü]\*

(?<='+ unvan + r')\w\*\s+[A-ZÇĞİÖŞÜ][a-zçğıöşü]\*

* This regex iterates over the predefined “unvan” list and checks for the:
  + Unvan + Name
  + Unvan + Name Surname

[A-ZÇĞİÖŞÜ][a-zçğıöşü]\*\s+' + suf + r'\w\*'

* This regex iterates over the predefined “suffix” list and checks for the:
  + Name + suffix

1. Locations:

[A-ZÇĞİÖŞÜ]\w+

* A basic name search for just Location.

(?:[A-ZÇĞİÖŞÜ][A-ZÇĞİÖŞÜa-zçğıöşü]\*\s+){1,4}[A-ZÇĞİÖŞÜ][A-ZÇĞİÖŞÜa-zçğıöşü]\*

* This regex check for the Location name up to 5 words location if needed.

[A-ZÇĞİÖŞÜ]\w+\s\*' + locsuf + r'\w\*\s\*'

* This regex iterates over the predefined “locationsuffix” list and checks for the:
  + Location + Suffix

1. Organizations:

(?:[A-ZÇĞİÖŞÜ][A-ZÇĞİÖŞÜa-zçğıöşü]\*\s+){1,8}[A-ZÇĞİÖŞÜ][A-ZÇĞİÖŞÜa-zçğıöşü]\*

* This regex checks for the Organization name up to 9 words if needed.

(?:[A-ZÇĞİÖŞÜ][A-ZÇĞİÖŞÜa-zçğıöşü]\*\s+){1,8}[A-ZÇĞİÖŞÜ][a-zçğıöşü]\*\s+' + suffix + r'\w\*'

* This regex iterates over “organizationsuffix” list and checks for the:
  + OrganizationName{1,9} + Suffix

1. Date & Time:

These regex’s is well-explained in the code therefore just copied from the code.

# General

result = re.findall(r'\d{1,2}[-,:/]\d{1,2}[-,:/]\d{2,4}',line)

for out in result:

going2print.append(out)

# General time (clock) --> XX:XX

result = re.findall(r'\d{1,2}[:.]\d{1,2}',line)

for out in result:

going2print.append(out)

# General time (clock) --> XX AM PM

result = re.findall(r'\d{1,2}\s\*[AP][M]', line)

for out in result:

going2print.append(out)

#finding days --> DAY\_NAME

for day in days:

result = re.findall(day,line)

for out in result:

# printFormat(lineNumber, tip, out)

going2print.append(out)

# finding months --> MONTH\_NAME

for month in months:

result = re.findall(month,line)

for out in result:

# printFormat(lineNumber, tip, out)

going2print.append(out)

# finding months & years--> MONTH\_NAME XXXX

for month in months:

result = re.findall(month+r'\s+\d{4}\s\*',line)

for out in result:

# printFormat(lineNumber, tip, out)

going2print.append(out)

# finding months --> DD MONTH\_NAME YYYY

for month in monthsUpperCase:

result = re.findall( r'\d{1,2} ' + month + r' \d{4}',line)

for out in result:

# printFormat(lineNumber, tip, out)

going2print.append(out)

# finding months --> DD MONTH\_NAME'XX

for month in monthsUpperCase:

result = re.findall( r'\d{1,2} ' + month + r'\'?\w\*',line)

for out in result:

# printFormat(lineNumber, tip, out)

going2print.append(out)

# finding years --> YYYY

result = re.findall(r'\d{4}\'?\w+', line)

for out in result:

# printFormat(lineNumber, tip, out)

going2print.append(out)

# finding years --> YYYY yıl

result = re.findall(r'\d{4}(?=\s+yıl\w+)',line)

for out in result:

# printFormat(lineNumber, tip, out)

going2print.append(out)

# finding years --> YY. yüzyıl

result = re.findall(r'\d{1,2}\. [Yy]üzyıl\w\*',line)

for out in result:

# printFormat(lineNumber, tip, out)

going2print.append(out)

# finding years --> XXXX-XXXX

result = re.findall(r'\d{4}[-/]\d{4}', line)

for out in result:

# printFormat(lineNumber, tip, out)

going2print.append(out)

# finding years --> XXXX

result = re.findall(r'\d{4}', line)

for out in result:

# printFormat(lineNumber, tip, out)

going2print.append(out)

# finding years --> MÖ X...

result = re.findall(r'MÖ \d+', line)

for out in result:

# printFormat(lineNumber, tip, out)

going2print.append(out)