# *CSE6060*

# *Statistical Natural Language Processing*

## NLP - தமிழ்

**Name : Kavianand G**

**Reg. No. : 19MAI0050**

**Date : 27 – June – 2020**

# *Indic NLP Library*

The library provides the following functionalities:

- Text Normalization
- Script Information
- Tokenization
- Word Segmentation
- Script Conversion
- Romanization
- Indicization
- Transliteration
- Translation

In [1]:

```
# The path to the local git repo for Indic NLP library
INDIC_NLP_LIB_HOME=r"C:\Users\kavianand\Documents\src\indic_nlp_library"

# The path to the local git repo for Indic NLP Resources
INDIC_NLP_RESOURCES=r"C:\Users\kavianand\Documents\src\indic_nlp_resources"
```

In [2]:

```
import sys
sys.path.append(r'{}\src'.format(INDIC_NLP_LIB_HOME))
```

In [3]:

```
from indicnlp import common
common.set_resources_path(INDIC_NLP_RESOURCES)
```

In [4]:

```
from indicnlp import loader
loader.load()
```

In [5]:

```
data="""தமிழ் இலக்கியம் இரண்டாயிரம் ஆண்டுகளுக்கு மேலான தொடர்ச்சி கொண்

print(data)
```

"தமிழ் இலக்கியம் இரண்டாயிரம் ஆண்டுகளுக்கு மேலான தொடர்ச்சி கொ
ண்ட உலகின் சிறந்த இலக்கியங்களில் ஒன்று. வாழ்வின் பல்வேறு கூறுக
ளை தமிழ் இலக்கியங்கள் இயம்புகின்றன. தமிழ் இலக்கியத்தில் வெ
ண்பா, குறள், புதுக்கவிதை, கட்டுரை, பழமொழி, தொண்ணூற்றாறு வ
கை சிற்றிலக்கியங்கள் என பல வடிவங்கள் உள்ளன. தமிழில் வாய்மொழி
இலக்கியங்களும் முக்கிய இடம் வகிக்கின்றன.

## Text Normalization

In [6]:

```
from indicnlp.normalize.indic_normalize import IndicNormalizerFactory

input_text= data

factory=IndicNormalizerFactory()
normalizer=factory.get_normalizer("ta")
output_text=normalizer.normalize(input_text)

print(input_text)
print()

print('Before normalization')
print(' '.join([ c for c in input_text ] ))
print('Length: {}'.format(len(input_text)))
print()
print('After normalization')
print(' '.join([ c for c in output_text ] ))
print('Length: {}'.format(len(output_text)))

```

"தமிழ் இலக்கியம் இரண்டாயிரம் ஆண்டுகளுக்கு மேலான தொடர்ச்சி கொண்ட உலகின் சிறந்த இலக்கியங்களில் ஒன்று. வாழ்வின் பல்வேறு கூறுகளை தமிழ் இலக்கியங்கள் இயம்புகின்றன. தமிழ் இலக்கியத்தில் வெண்பா, குறள், புதுக்கவிதை, கட்டுரை, பழமொழி, தொண்ணூற்றாறு வகை சிற்றிலக்கியங்கள் என பல வடிவங்கள் உள்ளன. தமிழில் வாய்மொழி இலக்கியங்களும் முக்கிய இடம் வகிக்கின்றன.

Before normalization
" த ம ி ழ ் இ ல க ் க ி ய ம ் இ ர ண ் ட ா ய ி ர ம ் ஆ ண ் ட ு க ள ு க ் க ு ம ே ல ா ன த ொ ட ர ் ச ் ச ி க ொ ண ் ட உ ல க ி ன ் ச ி ற ந ் த இ ல க ் க ி ய ங ் க ள ி ல ் ஒ ன ் ற ு . வ ா ழ ் வ ி ன ் ப ல ் வ ே ற ு க ூ ற ு க ள ை த ம ி ழ ் இ ல க ் க ி ய ங ் க ள ் இ ய ம ் ப ு க ி ன ் ற ன . த ம ி ழ ் இ ல க ் க ி ய த ் த ி ல ் வ ெ ண ் ப ா , க ு ற ள ் , ப ு த ு க ் க வ ி த ை , க ட ் ட ு ர ை , ப ழ ம ொ ழ ி , த ொ ண ் ண ூ ற ் ற ா ற ு வ க ை ச ி ற ் ற ி ல க ் க ி ய ங ் க ள ் எ ன ப ல வ ட ி வ ங ் க ள ் உ ள ் ள ன . த ம ி ழ ி ல ் வ ா ய ் ம ொ ழ ி இ ல க ் க ி ய ங ் க ள ு ம ் ம ு க ் க ி ய இ ட ம ் வ க ி க ் க ி ன ் ற ன .
Length: 338

After normalization
" த ம ி ழ ் இ ல க ் க ி ய ம ் இ ர ண ் ட ா ய ி ர ம ் ஆ ண ் ட ு க ள ு க ் க ு ம ே ல ா ன த ொ ட ர ் ச ் ச ி க ொ ண ் ட உ ல க ி ன ் ச ி ற ந ் த இ ல க ் க ி ய ங ் க ள ி ல ் ஒ ன ் ற ு . வ ா ழ ் வ ி ன ் ப ல ் வ ே ற ு க ூ ற ு க ள ை த ம ி ழ ் இ ல க ் க ி ய ங ் க ள ் இ ய ம ் ப ு க ி ன ் ற ன . த ம ி ழ ் இ ல க ் க ி ய த ் த ி ல ் வ ெ ண ் ப ா , க ு ற ள ் , ப ு த ு க ் க வ ி த ை , க ட ் ட ு ர ை , ப ழ ம ொ ழ ி , த ொ ண ் ண ூ ற ் ற ா ற ு வ க ை ச ி ற ் ற ி ல க ் க ி ய ங ் க ள ் எ ன ப ல வ ட ி வ ங ் க ள ் உ ள ் ள ன . த ம ி ழ ி ல ் வ ா ய ் ம ொ ழ ி இ ல க ் க ி ய ங ் க ள ு ம ் ம ு க ் க ி ய இ ட ம ் வ க ி க ் க ி ன ் ற ன .
Length: 338

# Sentence Tokenization / Sentence Splitter

In [7]:

```
from indicnlp.tokenize import sentence_tokenize

string = data
sentences=sentence_tokenize.sentence_split(string, lang='ta')
for t in sentences:
    print("\n",t)
```

"தமிழ் இலக்கியம் இரண்டாயிரம் ஆண்டுகளுக்கு மேலான தொடர்ச்சி கொண்ட உலகின் சிறந்த இலக்கியங்களில் ஒன்று.

வாழ்வின் பல்வேறு கூறுகளை தமிழ் இலக்கியங்கள் இயம்புகின்றன.

தமிழ் இலக்கியத்தில் வெண்பா, குறள், புதுக்கவிதை, கட்டுரை, பழமொழி, தொண்ணூற்றாறு வகை சிற்றிலக்கியங்கள் என பல வடிவங்கள் உள்ளன.

தமிழில் வாய்மொழி இலக்கியங்களும் முக்கிய இடம் வகிக்கின்றன.

# Word Tokenization

In [8]:

```
from indicnlp.tokenize import indic_tokenize
from indicnlp.tokenize import indic_detokenize

string=data

print('\nInput String: {}'.format(string))
print('\nTokens: ')
for t in indic_tokenize.trivial_tokenize(string):
    print(t)
```

```
Input String: "தமிழ் இலக்கியம் இரண்டாயிரம் ஆண்டுகளுக்கு மேலான
தொடர்ச்சி கொண்ட உலகின் சிறந்த இலக்கியங்களில் ஒன்று. வாழ்வின் ப
ல்வேறு கூறுகளை தமிழ் இலக்கியங்கள் இயம்புகின்றன. தமிழ் இலக்கிய
த்தில் வெண்பா, குறள், புதுக்கவிதை, கட்டுரை, பழமொழி, தொ
ண்ணூற்றாறு வகை சிற்றிலக்கியங்கள் என பல வடிவங்கள் உள்ளன. த
மிழில் வாய்மொழி இலக்கியங்களும் முக்கிய இடம் வகிக்கின்றன.

Tokens:
"
தமிழ்
இலக்கியம்
இரண்டாயிரம்
ஆண்டுகளுக்கு
மேலான
தொடர்ச்சி
கொண்ட
உலகின்
சிறந்த
இலக்கியங்களில்
ஒன்று
.
வாழ்வின்
பல்வேறு
கூறுகளை
தமிழ்
இலக்கியங்கள்
இயம்புகின்றன
.
தமிழ்
இலக்கியத்தில்
வெண்பா
,
குறள்
,
புதுக்கவிதை
,
கட்டுரை
,
பழமொழி
,
தொண்ணூற்றாறு
வகை
சிற்றிலக்கியங்கள்
என
பல
வடிவங்கள்
உள்ளன
.
```

தமிழில்
வாய்மொழி
இலக்கியங்களும்
முக்கிய
இடம்
வகிக்கின்றன
.

## Syllabification

In [9]:

```
from indicnlp.syllable import  syllabifier

text ='சிட்டுக்குருவி மனித குடியிருப்புடன் வலுவாகத் தொடர்புடையது ஆகும்ி'
lang='ta'

syl_word = (' '.join(syllabifier.orthographic_syllabify(text,lang)))
print("Syllabification (Approximation)\n")
print(syl_word)## Transliteration
```

```
Syllabification (Approximation)

சி ட்டு க்கு ரு வி  ம னி த  கு டி யி ரு ப்பு ட ன்  வ லு வா க த்  தொ
ட ர்பு டை ய து  ஆ கு ம்ி
```

## Morphological Analyser

In [10]:

```
from indicnlp.morph import unsupervised_morph
from indicnlp import common

analyzer=unsupervised_morph.UnsupervisedMorphAnalyzer('ta')## Syllabification
```

In [11]:

```
string = 'புத்துணர்ச்சியான'

analyzes_tokens = analyzer.morph_analyze_document(string.split(' '))
print(analyzes_tokens)
```

```
['புத்துணர்ச்சி', 'யான']
```

In [12]:

```
text ='சிட்டுக்குருவி மனித குடியிருப்புடன் வலுவாகத் தொடர்புடையது ஆகும்ி'

analyzes_tokens=analyzer.morph_analyze_document(text.split(' '))

for w in analyzes_tokens:
    print(w)
```

சிட்டுக்குருவி
மனித
குடியிருப்பு
டன்
வலுவாக
த்
தொடர்புடைய
து
ஆகும்
ி

# POS Tagging - தமிழ்

In [13]:

```
from rippletagger.tagger import Tagger

def pos_tag_tamil(query):
    tagger = Tagger(language="tam")
    posTagger = tagger.tag(query)
    return(posTagger)
```

In [14]:

```
query ='சிட்டுக்குருவி மனித குடியிருப்புடன் வலுவாகத் தொடர்புடையது ஆகும்ி'
tamil_tag = pos_tag_tamil(query)
print(tamil_tag)
```

```
[('சிட்டுக்குருவி', 'NOUN'), ('மனித', 'PROPN'), ('குடியிருப்புடன்', 'NOUN'),
('வலுவாகத்', 'PROPN'), ('தொடர்புடையது', 'VERB'), ('ஆகும்ி', 'VERB')]
```

In [15]:

```
query = "தமிழ் இலக்கியம் இரண்டாயிரம் ஆண்டுகளுக்கு மேலான தொடர்ச்சி கொண்
tamil_tag = pos_tag_tamil(query)
print(tamil_tag)
```

```
[('தமிழ்', 'PROPN'), ('இலக்கியம்', 'NOUN'), ('இரண்டாயிரம்', 'NOUN'),
('ஆண்டுகளுக்கு', 'NOUN'), ('மேலான', 'ADJ'), ('தொடர்ச்சி', 'NOUN'),
('கொண்ட', 'ADP'), ('உலகின்', 'NOUN'), ('சிறந்த', 'ADJ'), ('இலக்கிய
ங்களில்', 'NOUN'), ('ஒன்று.', 'NOUN'), ('வாழ்வின்', 'NOUN'), ('பல்வேறு',
'ADJ'), ('கூறுகளை', 'NOUN'), ('தமிழ்', 'PROPN'), ('இலக்கியங்கள்', 'NOU
N'), ('இயம்புகின்றன.', 'NOUN'), ('தமிழ்', 'PROPN'), ('இலக்கியத்தில்',
'NOUN'), ('வெண்பா,', 'NOUN'), ('குறள்,', 'NOUN'), ('புதுக்கவிதை,', 'NOU
N'), ('கட்டுரை,', 'NOUN'), ('பழமொழி,', 'NOUN'), ('தொண்ணூற்றாறு',
'NOUN'), ('வகை', 'NOUN'), ('சிற்றிலக்கியங்கள்', 'NOUN'), ('என', 'PAR
T'), ('பல', 'ADJ'), ('வடிவங்கள்', 'NOUN'), ('உள்ளன.', 'NOUN'), ('த
மிழில்', 'NOUN'), ('வாய்மொழி', 'NOUN'), ('இலக்கியங்களும்', 'ADJ'),
('முக்கிய', 'ADJ'), ('இடம்', 'ADP'), ('வகிக்கின்றன.', 'VERB')]
```

## StopWords - தமிழ்

In [16]:

```
with open('TamilStopWords.txt', encoding="utf8") as file:
    stop_words_tamil = file.read()

print(stop_words_tamil)
```

```
ஒரு
என்று
மற்றும்
இந்த
இது
என்ற
கொண்டு
என்பது
பல
ஆகும்
அல்லது
அவர்
நான்
உள்ள
அந்த
இவர்
என
முதல்
என்ன
```

In [17]:

```
data="""தமிழ் இலக்கியம் இரண்டாயிரம் ஆண்டுகளுக்கு மேலான தொடர்ச்சி கொண்

```

In [18]:

```
token = []
for t in indic_tokenize.trivial_tokenize(data):
    token.append(t)
```

In [19]:

```
print("Tokenized Words :\n ")
print(token)
```

```
Tokenized Words :

['"', 'தமிழ்', 'இலக்கியம்', 'இரண்டாயிரம்', 'ஆண்டுகளுக்கு', 'மே
லான', 'தொடர்ச்சி', 'கொண்ட', 'உலகின்', 'சிறந்த', 'இலக்கியங்க
ளில்', 'ஒன்று', '.', 'வாழ்வின்', 'பல்வேறு', 'கூறுகளை', 'தமிழ்', 'இல
க்கியங்கள்', 'இயம்புகின்றன', '.', 'தமிழ்', 'இலக்கியத்தில்', 'வெ
ண்பா', ',', 'குறள்', ',', 'புதுக்கவிதை', ',', 'கட்டுரை', ',', 'பழ
மொழி', ',', 'தொண்ணூற்றாறு', 'வகை', 'சிற்றிலக்கியங்கள்', 'என',
'பல', 'வடிவங்கள்', 'உள்ளன', '.', 'தமிழில்', 'வாய்மொழி', 'இலக்கிய
ங்களும்', 'முக்கிய', 'இடம்', 'வகிக்கின்றன', '.']
```

In [20]:

```
fil_sen = []
fil_sw = []

for t in indic_tokenize.trivial_tokenize(data):
    if t not in stop_words_tamil:
        fil_sen.append(t)
    else:
        fil_sw.append(t)
```

In [21]:

```
print(fil_sen)
```

```
['"', 'தமிழ்', 'இலக்கியம்', 'இரண்டாயிரம்', 'ஆண்டுகளுக்கு', 'மே
லான', 'தொடர்ச்சி', 'உலகின்', 'சிறந்த', 'இலக்கியங்களில்', 'ஒன்று',
'.', 'வாழ்வின்', 'கூறுகளை', 'தமிழ்', 'இலக்கியங்கள்', 'இயம்புகின்ற
ன', '.', 'தமிழ்', 'இலக்கியத்தில்', 'வெண்பா', ',', 'குறள்', ',', 'பு
துக்கவிதை', ',', 'கட்டுரை', ',', 'பழமொழி', ',', 'தொண்ணூற்றாறு',
'வகை', 'சிற்றிலக்கியங்கள்', 'வடிவங்கள்', '.', 'தமிழில்', 'வா
ய்மொழி', 'இலக்கியங்களும்', 'முக்கிய', 'வகிக்கின்றன', '.']
```

In [22]:

```
print("Removed Stop Words :\n ")
print(fil_sw)
```

```
Removed Stop Words :

['கொண்ட', 'பல்வேறு', 'என', 'பல', 'உள்ளன', 'இடம்']
```

In [23]:

```
print("Number of Words including StopWords : " ,len(token))
print("Number of Words excluding StopWords : " ,len(fil_sen))
print("Total number of Stop Words Excluded : " , len(fil_sw))
```

```
Number of Words including StopWords :  47
Number of Words excluding StopWords :  41
Total number of Stop Words Excluded :  6
```

# Transliteration

In [24]:

```
from indicnlp.transliterate.unicode_transliterate import UnicodeIndicTransliterator

input_text = 'சிட்டுக்குருவி மனித குடியிருப்புடன் வலுவாகத் தொடர்புடையது ஆகும்'
ta_hi_text = (UnicodeIndicTransliterator.transliterate(input_text,"ta","hi"))

print("Transliterate From Tamil to Hindi\n")
print("Input Text : " , input_text)
print("output Text : ", ta_hi_text)
```

```
Transliterate From Tamil to Hindi

Input Text :  சிட்டுக்குருவி மனித குடியிருப்புடன் வலுவாகத் தொடர்புடைய
து ஆகும்
output Text :  चिट्टुक्कुरुवि मनित कुटियिरुप्पुटन् वलुवाकत् तॊटर्पुटैयतु आकुम्
```

In [25]:

```
from indicnlp.transliterate.unicode_transliterate import UnicodeIndicTransliterator

input_text=ta_hi_text
hi_ta_text = (UnicodeIndicTransliterator.transliterate(input_text,"hi","ta"))

print("Transliterate From Hindi to Tamil\n")
print("Input Text : " , input_text)
print("output Text : ", hi_ta_text)
```

```
Transliterate From Hindi to Tamil

Input Text :  चिट्टुक्कुरुवि मनित कुटियिरुप्पुटन् वलुवाकत् तॊटर्पुटैयतु आकुम्
output Text :  சிட்டுக்குருவி மனித குடியிருப்புடன் வலுவாகத் தொட
ர்புடையது ஆகும்
```

In [26]:

```
from indicnlp.transliterate.unicode_transliterate import ItransTransliterator

input_text = 'சிட்டுக்குருவி மனித குடியிருப்புடன் வலுவாகத் தொடர்புடையது ஆகும்'
lang='ta'

print(ItransTransliterator.to_itrans(input_text,lang))

print("Transliterate From Hindi to Tamil\n")
print("Input Text : " , input_text)
print("output Text : ", hi_ta_text)
```

```
chiTTukkuruvi ma*nita kuTiyiruppuTa*n valuvaakat t.oTarpuTaiyatu aakum
Transliterate From Hindi to Tamil

Input Text :  சிட்டுக்குருவி மனித குடியிருப்புடன் வலுவாகத் தொடர்புடைய
து ஆகும்
output Text :  சிட்டுக்குருவி மனித குடியிருப்புடன் வலுவாகத் தொட
ர்புடையது ஆகும்
```

## Translation Options

In [27]:

```
text ='சிட்டுக்குருவி மனித குடியிருப்புடன் வலுவாகத் தொடர்புடையது ஆகும்ி'## St
```

In [28]:

```
from textblob import TextBlob
txt = TextBlob(text)
# Can detect language
txt.detect_language()
```

Out[28]:

```
'ta'
```

In [29]:

```
# Language Translation also available in TextBlob
ta_eng_text = txt.translate(from_lang='ta', to ='en')
print(ta_eng_text)
```

```
The sparrow is strongly associated with human habitation
```

In [30]:

```
txt = ta_eng_text
# Language Translation also available in TextBlob
print(txt.translate(from_lang='en', to ='ta'))
```

குருவி மனித வாழ்விடத்துடன் வலுவாக தொடர்புடையது