Medical Text Preprocessing using NLP

Aim

To apply preprocessing techniques to sensitive medical or healthcare-related text using NLP tools.

```
medical_text = """"The patient is a 54-year-old male with a history of controlled hypertension and seasonal asthma who prese
"""
```

```
import nltk
nltk.download('punkt')
nltk.download('punkt_tab') # Added to resolve LookupError
from nltk.tokenize import word_tokenize
word_tokenize(medical_text)
```

```
 'an',
 'echocardiogram',
 'revealed',
 'preserved',
 'left',
 'ventricular',
 'function',
 'without',
 'significant',
 'valvular',
 'abnormalities',
 '.',
 'The',
 'attending',
 'physician',
 'recommended',
 'lifestyle',
 'modification',
 'including',
 'a',
 'heart-healthy',
 'diet',
 ',',
 'increased',
 'physical',
 'activity',
 ',',
 'and',
 'stress',
 'management',
 ',',
 'along',
 'with',
 'initiating',
 'a',
 'low-dose',
 'statin',
 'and',
 'scheduling',
 'a',
 'follow-up',
 'stress',
 'test',
 'in',
 'four',
 'weeks',
 'to',
 'assess',
 'for',
 'potential',
 'ischemic',
 'changes',
 'and',
 'refine',
 'the',
 'treatment',
 'plan',
 '.']
```

TOKENIZING BY SENTENCE

```
from nltk.tokenize import sent_tokenize
sent_tokenize(medical_text)
```

```
['"The patient is a 54-year-old male with a history of controlled hypertension and seasonal asthma who presented to the
outpatient clinic with complaints of persistent fatigue, intermittent chest discomfort, and occasional palpitations over
the past three weeks.',
 'On physical examination, his blood pressure was 142/88 mmHg with a regular pulse of 86 bpm and normal respiratory
auscultation.',
 'Baseline laboratory tests indicated mildly elevated cholesterol levels and slightly increased C-reactive protein,
prompting further cardiac evaluation.',
 'An electrocardiogram showed non-specific T-wave changes, and an echocardiogram revealed preserved left ventricular
function without significant valvular abnormalities.',
 'The attending physician recommended lifestyle modification including a heart-healthy diet, increased physical activity,
and stress management, along with initiating a low-dose statin and scheduling a follow-up stress test in four weeks to
assess for potential ischemic changes and refine the treatment plan.']
```

FILTERING STOP WORDS

```
nltk.download("stopwords")
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
words_in_quote = word_tokenize(medical_text)
words_in_quote
```

```
 'an',
 'echocardiogram',
 'revealed',
 'preserved',
 'left',
 'ventricular',
 'function',
 'without',
 'significant',
 'valvular',
 'abnormalities',
 '.',
 'The',
 'attending',
 'physician',
 'recommended',
 'lifestyle',
 'modification',
 'including',
 'a',
 'heart-healthy',
 'diet',
 ',',
 'increased',
 'physical',
 'activity',
 ',',
 'and',
 'stress',
 'management',
 ',',
 'along',
 'with',
 'initiating',
 'a',
 'low-dose',
 'statin',
 'and',
 'scheduling',
 'a',
 'follow-up',
 'stress',
 'test',
 'in',
 'four',
 'weeks',
 'to',
 'assess',
 'for',
 'potential',
 'ischemic',
 'changes',
 'and',
 'refine',
 'the',
 'treatment',
 'plan',
 '.']
```

```
stop_words = set(stopwords.words("english"))
filtered_list = []
for word in words_in_quote:
  if word.casefold() not in stop_words:
    filtered_list.append(word)
filtered_list
```

```
 'C-reactive',
 'protein',
 ',',
 'prompting',
 'cardiac',
 'evaluation',
 '.',
 'electrocardiogram',
 'showed',
 'non-specific',
 'T-wave',
 'changes',
 ',',
 'echocardiogram',
 'revealed',
 'preserved',
 'left',
 'ventricular',
 'function',
 'without',
 'significant',
 'valvular',
 'abnormalities',
 '.',
 'attending',
 'physician',
 'recommended',
 'lifestyle',
 'modification',
 'including',
 'heart-healthy',
 'diet',
 ',',
 'increased',
 'physical',
 'activity',
 ',',
 'stress',
 'management',
 ',',
 'along',
 'initiating',
 'low-dose',
 'statin',
 'scheduling',
 'follow-up',
 'stress',
 'test',
 'four',
 'weeks',
 'assess',
 'potential',
 'ischemic',
 'changes',
 'refine',
 'treatment',
 'plan',
 '.']
```

STEMMING

```
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
stemmer = PorterStemmer()
words = word_tokenize(medical_text)
stemmed_words = [stemmer.stem(word) for word in words]
stemmed_words
```

```
 'the'
 'the',
 'attend',
 'physician',
 'recommend',
 'lifestyl',
 'modif',
 'includ',
 'a',
 'heart-healthi',
 'diet',
 ',',
 'increas',
 'physic',
 'activ',
 ',',
 'and',
 'stress',
 'manag',
 ',',
 'along',
 'with',
 'initi',
 'a',
 'low-dos',
 'statin',
 'and',
 'schedul',
 'a',
 'follow-up',
 'stress',
 'test',
 'in',
 'four',
 'week',
 'to',
 'assess',
 'for',
 'potenti',
 'ischem',
 'chang',
 'and',
 'refin',
 'the',
 'treatment',
 'plan',
```

UNDERSTEMMING AND OVERSTEMMING

```
from nltk.stem import SnowballStemmer
snowball = SnowballStemmer(language='english')
words = word_tokenize(medical_text)
for word in words:
    print(word,"--->",snowball.stem(word))
```

```
`` ---> ``
The ---> the
patient ---> patient
is ---> is
a ---> a
54-year-old ---> 54-year-old
male ---> male
with ---> with
a ---> a
history ---> histori
of ---> of
controlled ---> control
hypertension ---> hypertens
and ---> and
seasonal ---> season
asthma ---> asthma
who ---> who
presented ---> present
to ---> to
the ---> the
outpatient ---> outpati
clinic ---> clinic
with ---> with
complaints ---> complaint
of ---> of
persistent ---> persist
fatigue ---> fatigu
, ---> ,
intermittent ---> intermitt
chest ---> chest
discomfort ---> discomfort
, ---> ,
and ---> and
occasional ---> occasion
```

```
palpitations ---> palpit
over ---> over
the ---> the
past ---> past
three ---> three
weeks ---> week
. ---> .
On ---> on
physical ---> physic
examination ---> examin
, ---> ,
his ---> his
blood ---> blood
pressure ---> pressur
was ---> was
142/88 ---> 142/88
mmHg ---> mmhg
with ---> with
a ---> a
regular ---> regular
pulse ---> puls
of ---> of
86 ---> 86
bpm ---> bpm
```

```
from nltk import LancasterStemmer
Lanc = LancasterStemmer()
words = word_tokenize(medical_text)
for word in words:
    print(word,"--->",Lanc.stem(word))
```

```
an ---> an
echocardiogram ---> echocardiogram
revealed ---> rev
preserved ---> preserv
left ---> left
ventricular ---> ventricul
function ---> funct
without ---> without
significant ---> sign
valvular ---> valvul
abnormalities ---> abnorm
. ---> .
The ---> the
attending ---> attend
physician ---> phys
recommended ---> recommend
lifestyle ---> lifestyl
modification ---> mod
including ---> includ
a ---> a
heart-healthy ---> heart-healthy
diet ---> diet
, ---> ,
increased ---> increas
physical ---> phys
activity ---> act
, ---> ,
and ---> and
stress ---> stress
management ---> man
, ---> ,
along ---> along
with ---> with
initiating ---> in
a ---> a
low-dose ---> low-dose
statin ---> statin
and ---> and
scheduling ---> scheduling
a ---> a
follow-up ---> follow-up
stress ---> stress
test ---> test
in ---> in
four ---> four
weeks ---> week
to ---> to
assess ---> assess
for ---> for
potential ---> pot
ischemic ---> ischem
changes ---> chang
and ---> and
refine ---> refin
the ---> the
treatment ---> tre
plan ---> plan
. ---> .
```

```
from nltk.stem import RegexpStemmer
regexp = RegexpStemmer('ing|s|e|able', min=4)
words = word_tokenize(medical_text)
for word in words:
    print(word,"--->",Lanc.stem(word))
```

```
`` ---> ``
The ---> the
patient ---> paty
is ---> is
a ---> a
54-year-old ---> 54-year-old
male ---> mal
with ---> with
a ---> a
history ---> hist
of ---> of
controlled ---> control
hypertension ---> hypertend
and ---> and
seasonal ---> season
asthma ---> asthm
who ---> who
presented ---> pres
to ---> to
the ---> the
outpatient ---> outpaty
clinic ---> clin
with ---> with
complaints ---> complaint
of ---> of
persistent ---> persist
fatigue ---> fatigu
, ---> ,
intermittent ---> intermit
chest ---> chest
discomfort ---> discomfort
, ---> ,
and ---> and
occasional ---> occas
palpitations ---> palpit
over ---> ov
the ---> the
past ---> past
three ---> three
weeks ---> week
. ---> .
On ---> on
physical ---> phys
examination ---> examin
, ---> ,
his ---> his
blood ---> blood
pressure ---> press
was ---> was
142/88 ---> 142/88
mmHg ---> mmhg
with ---> with
a ---> a
regular ---> regul
pulse ---> puls
of ---> of
86 ---> 86
bpm ---> bpm
```

LEMMATIZATION

```
nltk.download('omw-1.4')
nltk.download('wordnet') # Added to resolve LookupError
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
words = word_tokenize(medical_text)
for word in words:
    print(word,"--->",lemmatizer.lemmatize(word))
```

```
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
`` ---> ``
The ---> The
patient ---> patient
is ---> is
a ---> a
54-year-old ---> 54-year-old
male ---> male
with ---> with
a ---> a
```

```
history ---> history
of ---> of
controlled ---> controlled
hypertension ---> hypertension
and ---> and
seasonal ---> seasonal
asthma ---> asthma
who ---> who
presented ---> presented
to ---> to
the ---> the
outpatient ---> outpatient
clinic ---> clinic
with ---> with
complaints ---> complaint
of ---> of
persistent ---> persistent
fatigue ---> fatigue
, ---> ,
intermittent ---> intermittent
chest ---> chest
discomfort ---> discomfort
, ---> ,
and ---> and
occasional ---> occasional
palpitations ---> palpitation
over ---> over
the ---> the
past ---> past
three ---> three
weeks ---> week
. ---> .
On ---> On
physical ---> physical
examination ---> examination
, ---> ,
his ---> his
blood ---> blood
pressure ---> pressure
was ---> wa
142/88 ---> 142/88
mmHg ---> mmHg
with ---> with
a ---> a
regular ---> regular
```

```
lemmatizer.lemmatize("worst")
```

```
'worst'
```

```
lemmatizer.lemmatize("worst", pos="a")
```

```
'bad'
```

COMPARISON

```
from nltk.stem import PorterStemmer, SnowballStemmer, LancasterStemmer, RegexpStemmer, WordNetLemmatizer
porter = PorterStemmer()
lancaster = LancasterStemmer()
snowball = SnowballStemmer(language='english')
regexp = RegexpStemmer('ing|s|e|able', min=4)
lemmatizer = WordNetLemmatizer()

word_list = ["friend", "friendship", "friends", "friendships"]
print("{0:20}{1:20}{2:20}{3:30}{4:40}{5:50}".format("Word","Porter Stemmer","Snowball Stemmer","Lancaster Stemmer",'Regexp
for word in word_list:
    print("{0:20}{1:20}{2:20}{3:30}{4:40}{5:50}".format(word,porter.stem(word),snowball.stem(word),lancaster.stem(word),reg
```

```
Word                Porter Stemmer      Snowball Stemmer    Lancaster Stemmer             Regexp Stemmer
friend              friend              friend              friend                        frind
friendship          friendship          friendship          friend                        frindhip
friends             friend              friend              friend                        frind
friendships         friendship          friendship          friend                        frindhip
```

CONCLUSION

Word	Stem	NLTK Lemma	spaCy Lemma
presenting	present	presenting	present
examination	examin	examination	examination
elevated	elev	elevated	elevate

PPT QUESTION: Write preprocessing output for: “NLP models are transforming the world rapidly!” 1️⃣ Word tokens 2️⃣ Stemmed words 3️⃣ Lemmatized words

```
nltk.download('punkt_tab')
```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
True
```

```
import nltk
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer, WordNetLemmatizer
nltk.download('punkt')
nltk.download('punkt_tab')
nltk.download('wordnet')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
True
```

```
sentence = "NLP models are transforming the world rapidly!"
```

```
sentence = sentence.lower()
```

TOKENIZATION

```
tokens = word_tokenize(sentence)
print("Word Tokens:", tokens)
```

```
Word Tokens: ['nlp', 'models', 'are', 'transforming', 'the', 'world', 'rapidly', '!']
```

STEMMING

```
stemmer = PorterStemmer()
stemmed_words = [stemmer.stem(word) for word in tokens]
print("Stemmed Words:", stemmed_words)
```

```
Stemmed Words: ['nlp', 'model', 'are', 'transform', 'the', 'world', 'rapidli', '!']
```

LEMMATIZATION

```
lemmatizer = WordNetLemmatizer()
lemmatized_words = [lemmatizer.lemmatize(word) for word in tokens]
print("Lemmatized Words:", lemmatized_words)
```

```
Lemmatized Words: ['nlp', 'model', 'are', 'transforming', 'the', 'world', 'rapidly', '!']
```

```
import nltk
nltk.download('punkt')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
True
```

```
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer, WordNetLemmatizer

sentence = "NLP models are transforming the world rapidly!"

tokens = word_tokenize(sentence)
stemmed = [PorterStemmer().stem(w) for w in tokens]
lemmatized = [WordNetLemmatizer().lemmatize(w) for w in tokens]

print(tokens)
print(stemmed)
print(lemmatized)
```

```
['NLP', 'models', 'are', 'transforming', 'the', 'world', 'rapidly', '!']
['nlp', 'model', 'are', 'transform', 'the', 'world', 'rapidli', '!']
['NLP', 'model', 'are', 'transforming', 'the', 'world', 'rapidly', '!']
```

GITHUB QUESTION : Write preprocessing output for: text data “SRUniversity” 1️⃣ Word tokens 2️⃣ Stemmed words 3️⃣ Lemmatized words

Tokenizing

```
SRUniversity="""The SR University campus is located in Ananthasagar village of Hasanparthy Mandal in Warangal, Telangana, 
It is in 150 acres, with both separate hostel facilities for boys and girls.
There is a huge central library along with Indias largest Technology Business Incubator (TBI) in tier 2 cities."""
```

```
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize
word_tokenize(SRUniversity)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
['The',
 'SR',
 'University',
 'campus',
 'is',
 'located',
 'in',
 'Ananthasagar',
 'village',
 'of',
 'Hasanparthy',
 'Mandal',
 'in',
 'Warangal',
 ',',
 'Telangana',
 ',',
 'India',
 '.',
 'It',
 'is',
 'in',
 '150',
 'acres',
 ',',
 'with',
 'both',
 'separate',
 'hostel',
 'facilities',
 'for',
 'boys',
 'and',
 'girls',
 '.',
 'There',
 'is',
 'a',
 'huge',
 'central',
 'library',
 'along',
 'with',
 'Indias',
 'largest',
 'Technology',
 'Business',
 'Incubator',
 '(',
 'TBI',
 ')',
 'in',
 'tier',
 '2',
 'cities',
 '.']
```

TOKENIZING BY SENTENCE

```
from nltk.tokenize import sent_tokenize
sent_tokenize(SRUniversity)
```

```
['The SR University campus is located in Ananthasagar village of Hasanparthy Mandal in Warangal, Telangana, India.',
 'It is in 150 acres, with both separate hostel facilities for boys and girls.',
 'There is a huge central library along with Indias largest Technology Business Incubator (TBI) in tier 2 cities.']
```

FILTERING STOP WORDS

```
nltk.download("stopwords")
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
words_in_quote = word_tokenize(SRUniversity)
words_in_quote
```

```
['The',
 'SR',
 'University',
 'campus',
 'is',
 'located',
 'in',
 'Ananthasagar',
 'village',
 'of',
 'Hasanparthy',
 'Mandal',
 'in',
 'Warangal',
 ',',
 'Telangana',
 ',',
 'India',
 '.',
 'It',
 'is',
 'in',
 '150',
 'acres',
 ',',
 'with',
 'both',
 'separate',
 'hostel',
 'facilities',
 'for',
 'boys',
 'and',
 'girls',
 '.',
 'There',
 'is',
 'a',
 'huge',
 'central',
 'library',
 'along',
 'with',
 'Indias',
 'largest',
 'Technology',
 'Business',
 'Incubator',
 '(',
 'TBI',
 ')',
 'in',
 'tier',
 '2',
 'cities',
 '.']
```

```
stop_words = set(stopwords.words("english"))
filtered_list = []
for word in words_in_quote:
  if word.casefold() not in stop_words:
    filtered_list.append(word)
filtered_list
```

```
['SR',
 'University',
 'campus',
 'located',
 'Ananthasagar',
 'village',
 'Hasanparthy',
 'Mandal',
 'Warangal',
 ',',
 'Telangana',
 ',',
 'India',
 '.',
 '150',
 'acres',
 ',',
 'separate',
 'hostel',
 'facilities',
 'boys',
 'girls',
 '.',
 'huge',
 'central',
 'library',
 'along',
 'Indias',
 'largest',
 'Technology',
 'Business',
 'Incubator',
 '(',
 'TBI',
 ')',
 'tier',
 '2',
 'cities',
 '.']
```

STEMMING

```
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
stemmer = PorterStemmer()
words = word_tokenize(SRUniversity)
stemmed_words = [stemmer.stem(word) for word in words]
stemmed_words
```

```
['the',
 'sr',
 'univers',
 'campu',
 'is',
 'locat',
 'in',
 'ananthasagar',
 'villag',
 'of',
 'hasanparthi',
 'mandal',
 'in',
 'warang',
 ',',
 'telangana',
 ',',
 'india',
 '.',
 'it',
 'is',
 'in',
 '150',
 'acr',
 ',',
 'with',
 'both',
 'separ',
 'hostel',
 'facil',
 'for',
 'boy',
 'and',
 'girl',
 '.',
 'there',
 'is',
 'a',
 'huge',
 'central',
 'librari',
 'along',
 'with',
 'india',
 'largest',
 'technolog',
 'busi',
 'incub',
 '(',
 'tbi',
 ')',
 'in',
 'tier',
 '2',
 'citi',
 '.']
```

UNDERSTEMMING AND OVERSTEMMING

```
from nltk.stem import SnowballStemmer
snowball = SnowballStemmer(language='english')
words = word_tokenize(SRUniversity)
for word in words:
    print(word,"--->",snowball.stem(word))
```

```
The ---> the
SR ---> sr
University ---> univers
campus ---> campus
is ---> is
located ---> locat
in ---> in
Ananthasagar ---> ananthasagar
village ---> villag
of ---> of
Hasanparthy ---> hasanparthi
Mandal ---> mandal
in ---> in
Warangal ---> warang
, ---> ,
Telangana ---> telangana
, ---> ,
India ---> india
. ---> .
It ---> it
is ---> is
in ---> in
150 ---> 150
acres ---> acr
, ---> ,
with ---> with
both ---> both
separate ---> separ
hostel ---> hostel
facilities ---> facil
for ---> for
boys ---> boy
and ---> and
girls ---> girl
. ---> .
There ---> there
is ---> is
a ---> a
huge ---> huge
central ---> central
library ---> librari
along ---> along
with ---> with
Indias ---> india
largest ---> largest
Technology ---> technolog
Business ---> busi
Incubator ---> incub
( ---> (
TBI ---> tbi
) ---> )
in ---> in
tier ---> tier
2 ---> 2
cities ---> citi
. ---> .
```

```
from nltk import LancasterStemmer
Lanc = LancasterStemmer()
```

```
words = word_tokenize(SRUniversity)
for word in words:
    print(word,"--->",Lanc.stem(word))
```

```
The ---> the
SR ---> sr
University ---> univers
campus ---> camp
is ---> is
located ---> loc
in ---> in
Ananthasagar ---> ananthasag
village ---> vil
of ---> of
Hasanparthy ---> hasanparthy
Mandal ---> mand
in ---> in
Warangal ---> warang
, ---> ,
Telangana ---> telangan
, ---> ,
India ---> ind
. ---> .
It ---> it
is ---> is
in ---> in
150 ---> 150
acres ---> acr
, ---> ,
with ---> with
both ---> both
separate ---> sep
hostel ---> hostel
facilities ---> facil
for ---> for
boys ---> boy
and ---> and
girls ---> girl
. ---> .
There ---> ther
is ---> is
a ---> a
huge ---> hug
central ---> cent
library ---> libr
along ---> along
with ---> with
Indias ---> india
largest ---> largest
Technology ---> technolog
Business ---> busy
Incubator ---> incub
( ---> (
TBI ---> tbi
) ---> )
in ---> in
tier ---> tier
2 ---> 2
cities ---> city
. ---> .
```

```
from nltk.stem import RegexpStemmer
regexp = RegexpStemmer('ing|s|e|able', min=4)
words = word_tokenize(SRUniversity)
for word in words:
    print(word,"--->",Lanc.stem(word))
```

```
The ---> the
SR ---> sr
University ---> univers
campus ---> camp
is ---> is
located ---> loc
in ---> in
Ananthasagar ---> ananthasag
village ---> vil
of ---> of
Hasanparthy ---> hasanparthy
Mandal ---> mand
in ---> in
Warangal ---> warang
, ---> ,
Telangana ---> telangan
, ---> ,
India ---> ind
. ---> .
It ---> it
is ---> is
```

```
in ---> in
150 ---> 150
acres ---> acr
, ---> ,
with ---> with
both ---> both
separate ---> sep
hostel ---> hostel
facilities ---> facil
for ---> for
boys ---> boy
and ---> and
girls ---> girl
. ---> .
There ---> ther
is ---> is
a ---> a
huge ---> hug
central ---> cent
library ---> libr
along ---> along
with ---> with
Indias ---> india
largest ---> largest
Technology ---> technolog
Business ---> busy
Incubator ---> incub
( ---> (
TBI ---> tbi
) ---> )
in ---> in
tier ---> tier
2 ---> 2
cities ---> city
. ---> .
```

LEMMATIZATION

```
nltk.download('omw-1.4')
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
words = word_tokenize(SRUniversity)
for word in words:
    print(word,"--->",lemmatizer.lemmatize(word))
```

```
The ---> The
SR ---> SR
University ---> University
campus ---> campus
is ---> is
located ---> located
in ---> in
Ananthasagar ---> Ananthasagar
village ---> village
of ---> of
Hasanparthy ---> Hasanparthy
Mandal ---> Mandal
in ---> in
Warangal ---> Warangal
, ---> ,
Telangana ---> Telangana
, ---> ,
India ---> India
. ---> .
It ---> It
is ---> is
in ---> in
150 ---> 150
acres ---> acre
, ---> ,
with ---> with
both ---> both
separate ---> separate
hostel ---> hostel
facilities ---> facility
for ---> for
boys ---> boy
and ---> and
girls ---> girl
. ---> .
There ---> There
is ---> is
a ---> a
huge ---> huge
central ---> central
library ---> library
along ---> along
with ---> with
```

```
Indias ---> Indias
largest ---> largest
Technology ---> Technology
Business ---> Business
Incubator ---> Incubator
( ---> (
TBI ---> TBI
) ---> )
in ---> in
tier ---> tier
2 ---> 2
cities ---> city
. ---> .
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
```

```
lemmatizer.lemmatize("worst")
```

```
'worst'
```

```
lemmatizer.lemmatize("worst", pos="a")
```

```
'bad'
```

COMPARISON

```
from nltk.stem import PorterStemmer, SnowballStemmer, LancasterStemmer, RegexpStemmer, WordNetLemmatizer
porter = PorterStemmer()
lancaster = LancasterStemmer()
snowball = SnowballStemmer(language='english')
regexp = RegexpStemmer('ing|s|e|able', min=4)
lemmatizer = WordNetLemmatizer()

word_list = ["friend", "friendship", "friends", "friendships"]
print("{0:20}{1:20}{2:20}{3:30}{4:40}{5:50}".format("Word","Porter Stemmer","Snowball Stemmer","Lancaster Stemmer",'Regexp
for word in word_list:
    print("{0:20}{1:20}{2:20}{3:30}{4:40}{5:50}".format(word,porter.stem(word),snowball.stem(word),lancaster.stem(word),reg
```

```
Word                Porter Stemmer      Snowball Stemmer    Lancaster Stemmer             Regexp Stemmer
friend              friend              friend              friend                        frind
friendship          friendship          friendship          friend                        frindhip
friends             friend              friend              friend                        frind
friendships         friendship          friendship          friend                        frindhip
```