MEDICAL TEXT PROCESSING USING NLP

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

```
!pip install nltk spacy
!python -m spacy download en_core_web_sm
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages (3.9.1)
Requirement already satisfied: spacy in /usr/local/lib/python3.12/dist-packages (3.8.11)
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from nltk) (8.3.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from nltk) (1.5.3)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.12/dist-packages (from nltk) (2025.11.
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from nltk) (4.67.1)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-packages (from spa
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from spa
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packages (from spacy
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (8.3
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.5
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.1.6)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from spacy) (75.2.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (25.0)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic!
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from pydan
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydant
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from reques
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0
```

```
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from thinc<
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from weas
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages (from weasel
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->spacy)
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.2.1-
Collecting en-core-web-sm==3.8.0
  Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.8.0/en_core_web_sm-3
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 113.8 MB/s eta 0:00:00
✓ Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
⚠ Restart to reload dependencies
If you are in a Jupyter or Colab notebook, you may need to restart Python in
order to load all the package's dependencies. You can do this by selecting the
'Restart kernel' or 'Restart runtime' option.
```

MEDICAL TEXT CORPOUS Sentence tokenization using NLTK.

```
import nltk
import spacy
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.stem import PorterStemmer
```

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

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

SENTENCE TOKENIZATION USING NLTK

```
medical_text = """
Hypertension is a chronic medical condition characterized by elevated blood pressure.
```

```
Patients with persistent hypertension are at increased risk of cardiovascular disease,
stroke, and kidney failure. Lifestyle modifications such as reduced salt intake,
regular physical activity, and medication adherence are essential for effective management.
Early diagnosis and continuous monitoring significantly improve patient outcomes.
"""
```

WORD TOKENIZATION USING NLTK

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

```
 'chronic',
 'medical',
 'condition',
 'characterized',
 'by',
 'elevated',
 'blood',
 'pressure',
 '.',
 'Patients',
 'with',
 'persistent',
 'hypertension',
 'are',
 'at',
 'increased',
 'risk',
 'of',
 'cardiovascular',
 'disease',
 ',',
```

```
 'kidney',
 'failure',
 '.',
 'Lifestyle',
 'modifications',
 'such',
 'as',
 'reduced',
 'salt',
 'intake',
 ',',
 'regular',
 'physical',
 'activity',
 ',',
 'and',
 'medication',
 'adherence',
 'are',
 'essential',
 'for',
 'effective',
 'management',
 '.',
 'Early',
 'diagnosis',
 'and',
 'continuous',
 'monitoring',
 'significantly',
 'improve',
 'patient',
 'outcomes',
 '.']
```

```
import nltk
from nltk.tokenize import sent_tokenize
nltk.download('punkt_tab')
medical_text = """
Hypertension is a chronic medical condition characterized by elevated blood pressure.
Patients with persistent hypertension are at increased risk of cardiovascular disease,
stroke, and kidney failure. Lifestyle modifications such as reduced salt intake,
```

```
regular physical activity, and medication adherence are essential for effective management.
Early diagnosis and continuous monitoring significantly improve patient outcomes.
"""
sent_tokenize(medical_text)
```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
['\nHypertension is a chronic medical condition characterized by elevated blood pressure.',
 'Patients with persistent hypertension are at increased risk of cardiovascular disease,\nstroke, and kidney
failure.',
 'Lifestyle modifications such as reduced salt intake,\nregular physical activity, and medication adherence
are essential for effective management.',
 'Early diagnosis and continuous monitoring significantly improve patient outcomes.']
```

filtering stop words

```
from nltk.tokenize import word_tokenize
words_in_quote = word_tokenize(medical_text)
words_in_quote
```

```
['Hypertension',
 'is',
 'a',
 'chronic',
 'medical',
 'condition',
 'characterized',
 'by',
 'elevated',
 'blood',
 'pressure',
 '.',
 'Patients',
 'with',
 'persistent',
 'hypertension',
 'are',
 'at',
 'increased',
 'risk',
```

```
 'of',
 'cardiovascular',
 'disease',
 ',',
 'stroke',
 ',',
 'and',
 'kidney',
 'failure',
 '.',
 'Lifestyle',
 'modifications',
 'such',
 'as',
 'reduced',
 'salt',
 'intake',
 ',',
 'regular',
 'physical',
 'activity',
 ',',
 'and',
 'medication',
 'adherence',
 'are',
 'essential',
 'for',
 'effective',
 'management',
 '.',
 'Early',
 'diagnosis',
 'and',
 'continuous',
 'monitoring',
 'significantly',
 'improve',
```

```
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
```

```
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
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
['Hypertension',
 'chronic',
 'medical',
 'condition',
 'characterized',
 'elevated',
 'blood',
 'pressure',
 '.',
 'Patients',
 'persistent',
 'hypertension',
 'increased',
 'risk',
 'cardiovascular',
 'disease',
 ',',
 'stroke',
 ',',
 'kidney',
 'failure',
 '.',
 'Lifestyle',
 'modifications',
 'reduced',
 'salt',
 'intake',
 ',',
 'regular',
 'physical',
 'activity',
 ',',
 'medication',
```

```
 'adherence',
 'essential',
 'effective',
 'management',
 '.',
 'Early',
 'diagnosis',
 'continuous',
 'monitoring',
 'significantly',
 'improve',
 'patient',
 'outcomes',
 '.']
```

STEMMING USING NLTK-PORTER STEMMER

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

medical_text = """
Hypertension is a chronic medical condition characterized by elevated blood pressure.
Patients with persistent hypertension are at increased risk of cardiovascular disease,
stroke, and kidney failure. Lifestyle modifications such as reduced salt intake,
regular physical activity, and medication adherence are essential for effective management.
Early diagnosis and continuous monitoring significantly improve patient outcomes.
"""

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

```
 'chronic',
 'medic',
 'condit',
```

```
 'blood',
 'pressur',
 '.',
 'patient',
 'with',
 'persist',
 'hypertens',
 'are',
 'at',
 'increas',
 'risk',
 'of',
 'cardiovascular',
 'diseas',
 ',',
 'stroke',
 ',',
 'and',
 'kidney',
 'failur',
 '.',
 'lifestyl',
 'modif',
 'such',
 'as',
 'reduc',
 'salt',
 'intak',
 ',',
 'regular',
 'physic',
 'activ',
 ',',
 'and',
 'medic',
 'adher',
 'are',
 'essenti',
 'for',
```

```
 'earli',
 'diagnosi',
 'and',
 'continu',
 'monitor',
 'significantli',
 'improv',
 'patient',
 'outcom',
```

understemming

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

```
Hypertension ---> hypertens
is ---> is
a ---> a
chronic ---> chronic
medical ---> medic
condition ---> condit
characterized ---> character
by ---> by
elevated ---> elev
blood ---> blood
pressure ---> pressur
. ---> .
Patients ---> patient
with ---> with
persistent ---> persist
hypertension ---> hypertens
are ---> are
at ---> at
increased ---> increas
risk ---> risk
of ---> of
cardiovascular ---> cardiovascular
```

```
disease ---> diseas
, ---> ,
stroke ---> stroke
, ---> ,
and ---> and
kidney ---> kidney
failure ---> failur
. ---> .
Lifestyle ---> lifestyl
modifications ---> modif
such ---> such
as ---> as
reduced ---> reduc
salt ---> salt
intake ---> intak
, ---> ,
regular ---> regular
physical ---> physic
activity ---> activ
, ---> ,
and ---> and
medication ---> medic
adherence ---> adher
are ---> are
essential ---> essenti
for ---> for
effective ---> effect
management ---> manag
. ---> .
Early ---> earli
diagnosis ---> diagnosi
and ---> and
continuous ---> continu
monitoring ---> monitor
significantly ---> signific
```

LEMMATIZATION USING spaCy

```
import nltk
nltk.download('omw-1.4')
nltk.download('wordnet') # Added to download the WordNet corpus
```

```
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...
Hypertension ---> Hypertension
is ---> is
a ---> a
chronic ---> chronic
medical ---> medical
condition ---> condition
characterized ---> characterized
by ---> by
elevated ---> elevated
blood ---> blood
pressure ---> pressure
. ---> .
Patients ---> Patients
with ---> with
persistent ---> persistent
hypertension ---> hypertension
are ---> are
at ---> at
increased ---> increased
risk ---> risk
of ---> of
cardiovascular ---> cardiovascular
disease ---> disease
, ---> ,
stroke ---> stroke
, ---> ,
and ---> and
kidney ---> kidney
failure ---> failure
. ---> .
Lifestyle ---> Lifestyle
modifications ---> modification
such ---> such
```

```
as ---> a
reduced ---> reduced
salt ---> salt
intake ---> intake
, ---> ,
regular ---> regular
physical ---> physical
activity ---> activity
, ---> ,
and ---> and
medication ---> medication
adherence ---> adherence
are ---> are
essential ---> essential
for ---> for
effective ---> effective
management ---> management
. ---> .
Early ---> Early
diagnosis ---> diagnosis
and ---> and
continuous ---> continuous
```

COMPARISION : ORIGINALvs STEMMED vs LEMMATIZED

```
Start coding or generate with AI.
```

```
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
```

Comparison of Stemming and Lemmatization for `medical_text`

```
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer, SnowballStemmer, LancasterStemmer, RegexpStemmer, WordNetLemmatizer
import nltk
```

```
nltk.download('wordnet')
nltk.download('omw-1.4')

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

# Ensure medical_text is defined
medical_text = """
Hypertension is a chronic medical condition characterized by elevated blood pressure.
Patients with persistent hypertension are at increased risk of cardiovascular disease,
stroke, and kidney failure. Lifestyle modifications such as reduced salt intake,
regular physical activity, and medication adherence are essential for effective management.
Early diagnosis and continuous monitoring significantly improve patient outcomes.
"""

words = word_tokenize(medical_text)

print("{0:20}{1:20}{2:20}{3:30}{4:40}{5:50}".format("Original Word","Porter Stemmer","Snowball Stemmer","Lanca
print("-" * 170)

for word in words:
    # Only process alphabetic words for better comparison, ignoring punctuation
    if word.isalpha():
        stemmed_porter = porter.stem(word)
        stemmed_snowball = snowball.stem(word)
        stemmed_lancaster = lancaster.stem(word)
        stemmed_regexp = regexp.stem(word)
        lemmatized_word = lemmatizer.lemmatize(word)
        print("{0:20}{1:20}{2:20}{3:30}{4:40}{5:50}".format(word, stemmed_porter, stemmed_snowball, stemmed_la
```

```
Original Word       Porter Stemmer      Snowball Stemmer    Lancaster Stemmer             Regexp Stemmer
-----------------------------------------------------------------------------------------------------------
Hypertension        hypertens           hypertens           hypertend                     Hyprtnion
is                  is                  is                  is                            is
a                   a                   a                   a                             a
chronic             chronic             chronic             chronic                       chronic
```

medical	medic	medic	med	mdical
condition	condit	condit	condit	condition
characterized	character	character	charact	charactrizd
by	by	by	by	by
elevated	elev	elev	elev	lvatd
blood	blood	blood	blood	blood
pressure	pressur	pressur	press	prur
Patients	patient	patient	paty	Patint
with	with	with	with	with
persistent	persist	persist	persist	pritnt
hypertension	hypertens	hypertens	hypertend	hyprtnion
are	are	are	ar	are
at	at	at	at	at
increased	increas	increas	increas	incrad
risk	risk	risk	risk	rik
of	of	of	of	of
cardiovascular	cardiovascular	cardiovascular	cardiovascul	cardiovacular
disease	diseas	diseas	diseas	dia
stroke	stroke	stroke	stroke	trok
and	and	and	and	and
kidney	kidney	kidney	kidney	kidny
failure	failur	failur	fail	failur
Lifestyle	lifestyl	lifestyl	lifestyl	Liftyl
modifications	modif	modif	mod	modification
such	such	such	such	uch
as	as	as	as	as
reduced	reduc	reduc	reduc	rducd
salt	salt	salt	salt	alt
intake	intak	intak	intak	intak
regular	regular	regular	regul	rgular
physical	physic	physic	phys	phyical
activity	activ	activ	act	activity
and	and	and	and	and
medication	medic	medic	med	mdication
adherence	adher	adher	adh	adhrnc
are	are	are	ar	are
essential	essenti	essenti	ess	ntial
for	for	for	for	for
effective	effect	effect	effect	ffctiv
management	manag	manag	man	managmnt
Early	earli	earli	ear	Early
diagnosis	diagnosi	diagnosi	diagnos	diagnoi

```
and                 and                 and                 and                 and
continuous          continu             continu             continu             continuou
monitoring          monitor             monitor             monit               monitor
significantly       significantli       signific            sign                ignificantly
improve             improv              improv              improv              improv
patient             patient             patient             paty                patint
outcomes            outcom              outcom              outcom              outcom
[nltk_data] Downloading package wordnet to /root/nltk_data...
```

PPT QUESTION: write preprocessing output for: "NLP models are transforming the world rapidly!"

1. word token
2. stemmed words
3. lemmatized words

```
import nltk
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

# Download required resources (run once)
nltk.download('punkt')
nltk.download('punkt_tab')   # ✅ IMPORTANT FIX
nltk.download('wordnet')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[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] Downloading package omw-1.4 to /root/nltk_data...
True
```

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

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

WORD TOKEN

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

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

STEMMED WORDS

```
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', '!']
```

LEMMATIZED WORDS

```
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', '!']
```

GITHUB QUESTION :write preprocessing output text data sr university

1. word token
2. stemmed word
3. lemmatized word

```
SRUniversity="""The SR University campus is located in Ananthasagar village of Hasanparthy Mandal in Warangal,
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',
 '.']
```

```
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.']
```

```
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',
 '.']
```

```
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',
 '.']
```

```
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
. ---> .
```

```
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'
```

```
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 Stem
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.s
```

```
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
```