```
Medical_txt = """
Medications are administered to reduce complications such as neuropathy and cardiovascular diseases.
Early diagnosis improves treatment outcomes and reduces mortality rates.
"""
```

Download Packages

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

```
[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]   Unzipping tokenizers/punkt_tab.zip.
['Medications',
 'are',
 'administered',
 'to',
 'reduce',
 'complications',
 'such',
 'as',
 'neuropathy',
 'and',
 'cardiovascular',
 'diseases',
 '.',
 'Early',
 'diagnosis',
 'improves',
 'treatment',
 'outcomes',
 'and',
 'reduces',
 'mortality',
 'rates',
 '.']
```

Tokenizing

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

```
['\nMedications are administered to reduce complications such as neuropathy and cardiovascular diseases.',
 'Early diagnosis improves treatment outcomes and reduces mortality rates.']
```

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_txt)
words_in_quote
```

```
['Medications',
 'are',
 'administered',
 'to',
 'reduce',
 'complications',
 'such',
 'as',
 'neuropathy',
 'and',
 'cardiovascular',
 'diseases',
 '.',
```

```
 'Early',
 'diagnosis',
 'improves',
 'treatment',
 'outcomes',
 'and',
 'reduces',
 'mortality',
 'rates',
 '.']
```

```
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
```

```
['Medications',
 'administered',
 'reduce',
 'complications',
 'neuropathy',
 'cardiovascular',
 'diseases',
 '.',
 'Early',
 'diagnosis',
 'improves',
 'treatment',
 'outcomes',
 'reduces',
 'mortality',
 'rates',
 '.']
```

Stemming

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

```
['medic',
 'are',
 'administ',
 'to',
 'reduc',
 'complic',
 'such',
 'as',
 'neuropathi',
 'and',
 'cardiovascular',
 'diseas',
 '.',
 'earli',
 'diagnosi',
 'improv',
 'treatment',
 'outcom',
 'and',
 'reduc',
 'mortal',
 'rate',
 '.']
```

SnowballStemmer

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

```
Medications ---> medic
are ---> are
administered ---> administ
to ---> to
reduce ---> reduc
complications ---> complic
such ---> such
as ---> as
neuropathy ---> neuropathi
and ---> and
cardiovascular ---> cardiovascular
diseases ---> diseas
. ---> .
Early ---> earli
diagnosis ---> diagnosi
improves ---> improv
treatment ---> treatment
outcomes ---> outcom
and ---> and
reduces ---> reduc
mortality ---> mortal
rates ---> rate
. ---> .
```

LancasterStemmer

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

```
Medications ---> med
are ---> ar
administered ---> admin
to ---> to
reduce ---> reduc
complications ---> comply
such ---> such
as ---> as
neuropathy ---> neuropathy
and ---> and
cardiovascular ---> cardiovascul
diseases ---> diseas
. ---> .
Early ---> ear
diagnosis ---> diagnos
improves ---> improv
treatment ---> tre
outcomes ---> outcom
and ---> and
reduces ---> reduc
mortality ---> mort
rates ---> rat
. ---> .
```

Import Lemmatization

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

```
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data] Downloading package wordnet to /root/nltk_data...
Medications ---> Medications
are ---> are
administered ---> administered
to ---> to
reduce ---> reduce
complications ---> complication
such ---> such
as ---> a
neuropathy ---> neuropathy
and ---> and
```

```
cardiovascular ---> cardiovascular
diseases ---> disease
. ---> .
Early ---> Early
diagnosis ---> diagnosis
improves ---> improves
treatment ---> treatment
outcomes ---> outcome
and ---> and
reduces ---> reduces
mortality ---> mortality
rates ---> rate
. ---> .
```

Comparison

```
from nltk.stem import PorterStemmer, SnowballStemmer, LancasterStemmer, RegexpStemmer, WordNetLemmatizer
porter = PorterStemmer()
lancaster = LancasterStemmer()
snowball = SnowballStemmer(language='english')
lemmatizer = WordNetLemmatizer()

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

```
Word                Porter Stemmer      Snowball Stemmer    Lancaster Stemmer             WordNetLemmatizer
friend              friend              friend              friend                        friend
friendship          friendship          friendship          friend                        friendship
friends             friend              friend              friend                        friend
friendships         friendship          friendship          friend                        friendship
```

SR University

```
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."""
```

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

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab 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',



 'hostel'
```

```
 '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]   Package stopwords is already up-to-date!
```

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

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

```
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|e', 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 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.stem(word),regexp.
```

```
Word                Porter Stemmer      Snowball Stemmer    Lancaster Stemmer            Regexp Stemmer
friend              friend              friend              friend                       frind
friendship          friendship          friendship          friend                       frindship
friends             friend              friend              friend                       frinds
friendships         friendship          friendship          friend                       frindships
```

preprocessing output of "NLP models are transforming the world rapidly!"

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

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

```
[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!
['NLP', 'models', 'are', 'transforming', 'the', 'world', 'rapidly', '!']
```

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

```
['NLP models are transforming the world rapidly!']
```

```
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]   Package stopwords is already up-to-date!
```

```
words_in_quote = word_tokenize(txt)
words_in_quote
```

```
['NLP', 'models', 'are', 'transforming', 'the', 'world', 'rapidly', '!']
```

```
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
```

```
['NLP', 'models', 'transforming', 'world', 'rapidly', '!']
```

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

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

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

```
NLP ---> nlp
models ---> model
are ---> are
transforming ---> transform
the ---> the
world ---> world
rapidly ---> rapid
! ---> !
```

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

```
NLP ---> nlp
models ---> model
are ---> ar
transforming ---> transform
```

```
the ---> the
world ---> world
rapidly ---> rapid
! ---> !
```

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

```
NLP ---> nlp
models ---> model
are ---> ar
transforming ---> transform
the ---> the
world ---> world
rapidly ---> rapid
! ---> !
```

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

```
NLP ---> NLP
models ---> model
are ---> are
transforming ---> transforming
the ---> the
world ---> world
rapidly ---> rapidly
! ---> !
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
```