```
!pip install nltk spacy

import nltk

nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')

!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) (2025.11.3)
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 spacy) (3
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.13)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (8.3.10)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.5.2)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.1
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.4.3)
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.20
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) (2.32.
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-packages (from
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!=1.8,
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0-
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->spacy) (3.0.3
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.2.1->wease
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
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-3.8.0-p
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 20.1 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.
```

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

```
file_path = "/content/data.txt"

with open(file_path, "r") as file:
    medical_text = file.read()
```

```
medical_text
```

```
'Excision of limbal dermoids. We reviewed the clinical files of 10 patients who had undergone excision of unilateral
epibulbar limbal dermoids. Preoperatively, all of the affected eyes had worse visual acuity (P less than .02) and mor
e astigmatism (P less than .01) than the contralateral eyes. Postoperatively, every patient was cosmetically improve
d. Of the eight patients for whom both preoperative and postoperative visual acuity measurements had been obtained, i
n six it had changed minimally (less than or equal to 1 line), and in two it had improved (less than or equal to 2 li
nes). Surgical complications included persistent epithelial defects (40%) and peripheral corneal vascularization and
opacity (70%). These complications do not outweigh the cosmetic and visual benefits of dermoid excision in selected p
```

```
import nltk
nltk.download('punkt_tab')
sentences = sent_tokenize(medical_text)
sentences
```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
['Excision of limbal dermoids.',
 'We reviewed the clinical files of 10 patients who had undergone excision of unilateral epibulbar limbal dermoids.',
 'Preoperatively, all of the affected eyes had worse visual acuity (P less than .02) and more astigmatism (P less
than .01) than the contralateral eyes.',
 'Postoperatively, every patient was cosmetically improved.',
 'Of the eight patients for whom both preoperative and postoperative visual acuity measurements had been obtained, in
six it had changed minimally (less than or equal to 1 line), and in two it had improved (less than or equal to 2
lines).',
 'Surgical complications included persistent epithelial defects (40%) and peripheral corneal vascularization and
opacity (70%).',
 'These complications do not outweigh the cosmetic and visual benefits of dermoid excision in selected patients.',
 "Bell's palsy.",
 'A diagnosis of exclusion.',
 'In cases of acute unilateral facial weakness, a careful and systematic evaluation is necessary to identify the
cause.',
 "Idiopathic facial paralysis (Bell's palsy) is a diagnosis of exclusion.",
 'It is also the most common cause of unilateral facial weakness seen by primary care physicians.',
 'The most important aspect of initial treatment is eye protection.',
 'Administration of systemic oral corticosteroids may lessen severity and duration of symptoms.']
```

```
words = word_tokenize(medical_text)
words
```

```
'initial'
```

```
 'initial',
 'treatment',
 'is',
 'eye',
 'protection',
 '.',
 'Administration',
 'of',
 'systemic',
 'oral',
 'corticosteroids',
 'may',
 'lessen',
 'severity',
 'and',
 'duration',
 'of',
 'symptoms',
 '.']
```

```
stop_words = set(stopwords.words("english"))

filtered_words = []
for word in words:
    if word.casefold() not in stop_words:
        filtered_words.append(word)

filtered_words
```

```
 'severity'
```

```
 'severity',
 'duration',
 'symptoms',
 '.']
```

```
porter = PorterStemmer()
porter_stemmed = [porter.stem(word) for word in filtered_words]
porter_stemmed
```

```
 "'s",
 'palsi',
 '.',
 'diagnosi',
 'exclus',
 '.',
 'case',
 'acut',
 'unilater',
 'facial',
 'weak',
 ',',
 'care',
 'systemat',
 'evalu',
 'necessari',
 'identifi',
 'caus',
 '.',
 'idiopath',
 'facial',
 'paralysi',
 '(',
 'bell',
 "'s",
 'palsi',
 ')',
 'diagnosi',
 'exclus',
 '.',
 'also',
 'common',
 'caus',
 'unilater',
 'facial',
 'weak',
 'seen',
 'primari',
 'care',
 'physician',
 '.',
 'import',
 'aspect',
 'initi',
 'treatment',
 'eye',
 'protect',
 '.',
 'administr',
 'system',
 'oral',
 'corticosteroid',
 'may',
 'lessen',
 'sever',
 'durat',
 'symptom',
 '.']
```

```
lemmatizer = WordNetLemmatizer()

lemmatized_words = [lemmatizer.lemmatize(word) for word in filtered_words]
lemmatized_words
```

```
 'unilateral',
```

```
 'unilateral',
 'facial',
 'weakness',
 ',',
 'careful',
 'systematic',
 'evaluation',
 'necessary',
 'identify',
 'cause',
 '.',
 'Idiopathic',
 'facial',
 'paralysis',
 '(',
 'Bell',
 "'s",
 'palsy',
 ')',
 'diagnosis',
 'exclusion',
 '.',
 'also',
 'common',
 'cause',
 'unilateral',
 'facial',
 'weakness',
 'seen',
 'primary',
 'care',
 'physician',
 '.',
 'important',
 'aspect',
 'initial',
 'treatment',
 'eye',
 'protection',
 '.',
 'Administration',
 'systemic',
 'oral',
 'corticosteroid',
 'may',
 'lessen',
 'severity',
 'duration',
 'symptom',
 '.']
```

```
nlp = spacy.load("en_core_web_sm")
doc = nlp(medical_text)

spacy_tokens = [token.text for token in doc]
spacy_tokens
```

```
 'weakness'
```

```
 'weakness',
 'seen',
 'by',
 'primary',
 'care',
 'physicians',
 '.',
 'The',
 'most',
 'important',
 'aspect',
 'of',
 'initial',
 'treatment',
 'is',
 'eye',
 'protection',
 '.',
 'Administration',
 'of',
 'systemic',
 'oral',
 'corticosteroids',
 'may',
 'lessen',
 'severity',
 'and',
 'duration',
 'of',
 'symptoms',
 '.']
```

```
spacy_lemmas = [(token.text, token.lemma_) for token in doc if not token.is_stop]
spacy_lemmas
```

```
 ('Administration', 'administration'),
 ('systemic', 'systemic'),
 ('oral', 'oral'),
 ('corticosteroids', 'corticosteroid'),
 ('lessen', 'lessen'),
 ('severity', 'severity'),
 ('duration', 'duration'),
 ('symptoms', 'symptom'),
 ('.', '.')]
```

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

```
SRUniversity
```

```
'The SR University campus is located in Ananthasagar village of Hasanparthy Mandal in Warangal, Telangana, India. \nI
t is in 150 acres, with both separate hostel facilities for boys and girls. \nThere is a huge central library along w
```

```
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]   Unzipping tokenizers/punkt_tab.zip.
['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',


 'tier'
```

```
 '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|e', min=4)
words = word_tokenize(SRUniversity)
for word in words:
    print(word,"--->",regexp.stem(word))
```

```
The ---> The
SR ---> SR
University ---> Univrsity
campus ---> campus
is ---> is
located ---> locatd
in ---> in
Ananthasagar ---> Ananthasagar
village ---> villag
of ---> of
Hasanparthy ---> Hasanparthy
Mandal ---> Mandal
in ---> in
Warangal ---> Warangal
, ---> ,
Telangana ---> Tlangana
, ---> ,
India ---> India
. ---> .
It ---> It
is ---> is
in ---> in
150 ---> 150
acres ---> acrs
, ---> ,
with ---> with
both ---> both
separate ---> sparat
hostel ---> hostl
facilities ---> facilitis
for ---> for
boys ---> boys
and ---> and
girls ---> girls
. ---> .
There ---> Thr
is ---> is
a ---> a
huge ---> hug
central ---> cntral
library ---> library
along ---> along
with ---> with
Indias ---> Indias
largest ---> largst
Technology ---> Tchnology
Business ---> Businss
Incubator ---> Incubator
( ---> (
TBI ---> TBI
) ---> )
in ---> in
tier ---> tir
2 ---> 2
cities ---> citis
. ---> .
```

```
import nltk
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))
```

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

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

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

```
'worst'
```

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

```
'bad'
```

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

```
NLP
```

```
'NLP models are transforming the world rapidly'
```

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

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
['NLP', 'models', 'are', 'transforming', 'the', 'world', 'rapidly']
```

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

```
['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(NLP)
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(NLP)
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(NLP)
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(NLP)
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(NLP)
for word in words:
    print(word,"--->",regexp.stem(word))
```

```
NLP ---> NLP
models ---> modls
are ---> are
transforming ---> transform
the ---> the
world ---> world
rapidly ---> rapidly
```

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

```
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
words = word_tokenize(NLP)
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!
```

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

```
'worst'
```

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

```
'bad'
```