**1.Tokenization in NLP**

**import spacy**

**nlp=spacy.load('en\_core\_web\_sm')**

**doc=nlp(u"Tesla is looking at buying U.S. startup for $6 million. Hi I am Happy")**

**for token in doc:**

**print(token.text,token.pos\_,token.dep\_)**

**print(nlp.pipeline)**

**print(nlp.pipe\_names)**

**print(doc[0])**

**print(doc)**

**print(type(doc))**

**spacy.explain(' nsubj')**

**print(doc[4].lemma\_)**

**spacy.explain(doc[4].tag\_)**

**print(doc[7].text , doc[7].shape\_)**

**print(doc[4].is\_alpha)**

**print(doc[2].is\_stop)**

**for sent in doc.sents:**

**print(sent)**

**print(doc[0].is\_sent\_start)**

**----------**

**2.Stemming in NLP  
import nltk**

**from nltk.stem.porter import \***

**from nltk.stem.snowball import SnowballStemmer**

**p\_stemmer=PorterStemmer()**

**words=['run','runner','running','ran','runs','easily','fairly']**

**for s in words:**

**print(s, p\_stemmer.stem(s))**

**s\_stemmer=SnowballStemmer(language='english')**

**for s in words:**

**print(s,s\_stemmer.stem(s))**

**dict='consolingly'**

**print( p\_stemmer.stem(dict))**

**print( s\_stemmer.stem(dict))**

**phrase='I am meeting him tommorow at the meeting'**

**for w in phrase.split():**

**print(w ,p\_stemmer.stem(w))  
3.Lematization in NLP**

**#Note cannot run this code twice..will show error**

**import spacy**

**nlp=spacy.load('en\_core\_web\_sm')**

**print(nlp.Defaults.stop\_words)**

**print(len(nlp.Defaults.stop\_words))**

**print(nlp.vocab['mystery'].is\_stop)**

**print(nlp.vocab['even'].is\_stop)**

**nlp.Defaults.stop\_words.add('btw') #add stop word**

**nlp.vocab['btw'].is\_stop=True #optional**

**print(nlp.vocab['btw'].is\_stop)**

**print(len(nlp.Defaults.stop\_words))**

**nlp.Defaults.stop\_words.remove("show") # remove stop word**

**print(nlp.vocab['four'].is\_stop)**

**print(len(nlp.Defaults.stop\_words))**

**4.Stop Words in NLP  
import spacy**

**nlp=spacy.load('en\_core\_web\_sm')**

**d=nlp(u"I am a runner running in a race because I love to run since I ran today")**

**for t in d:**

**print(t.text,t.pos\_,t.lemma,t.lemma\_)**

**def show\_lemma(text):**

**for t in text:**

**print(t.text,t.pos\_,t.lemma,t.lemma\_)**

**show\_lemma(d)**

**-------------  
5.Visualization of Parts Of Speech in NLP  
import spacy**

**from spacy import displacy**

**from IPython.core.display import display**

**nlp=spacy.load('en\_core\_web\_sm')**

**d=nlp(u"The quick brown fox jumped over the lazy dog's back")**

**html=displacy.render(d,style='dep',page=True)**

**options={'distance':110,'compact':'True','color':'yellow','bg':'black','font':'Times'}**

**html=displacy.render(d,style='dep',options=options)**

**6.Word Sense Disambiguation in NLP**

**import nltk**

**from nltk.wsd import lesk**

**from nltk.tokenize import word\_tokenize**

**from nltk.corpus import wordnet**

**def dw(sent,aw):**

**t=word\_tokenize(sent)**

**sense=lesk(t,aw,'n')**

**return sense**

**sent=[**

**'He went to the bank to withdraw money',**

**'she sat on the river bank to enjoy the view']**

**aw='bank'**

**for s in sent:**

**sense=dw(s,aw)**

**if sense:**

**print(s)**

**print("Disambiguated sense:",sense.name())**

**print(sense.definition())**

**else:**

**print("no sense found")**

**7.Regularization in NLP  
import re**

**txt='the agent phone-number is 4085551234. Calling the agent Soon!'**

**t='408-555-1234'**

**match=re.search('phone',txt)**

**print(match.start())**

**print(match.end())**

**print(match.span())**

**print(re.findall("agent",txt))**

**print(re.search(r'\d+',txt).group())**

**print("Area code:", re.search(r'\d+',t).group())**

**print("Line number:",re.search(r'\d{4}+',t).group())**

**print(re.search(r"phone|email",txt))**

**print(re.findall(r'\S+er',txt))**

**print(' '.join(re.findall('[^!.]+',txt)))**

**print(re.findall(r'[\w]+-[\w]+',txt))**

**print(re.search(r'Call(ing|ed|er)',txt))**

**8.Machine Translation in NLP**

**!pip install indic-nlp-library**

**!pip install indic-transliteration**

**from indic\_transliteration import sanscript**

**from indic\_transliteration.sanscript import transliterate**

**from indicnlp.tokenize.indic\_tokenize import trivial\_tokenize**

**from indicnlp.tokenize.sentence\_tokenize import sentence\_split**

**from indicnlp.transliterate.unicode\_transliterate import UnicodeIndicTransliterator**

**txt="நான் மகிழ்ச்சியாக இருக்கிறேன்."**

**text="I am happy"**

**t=trivial\_tokenize(txt)**

**detokenized\_txt=' '.join(t)**

**print(txt)**

**print(t)**

**print(detokenized\_txt)**

**print(sentence\_split(txt,lang='ta'))**

**ct=UnicodeIndicTransliterator.transliterate(txt,"ta","hi")**

**print(ct)**

**tt=transliterate(text,sanscript.ITRANS,sanscript.TAMIL)**

**print(tt)**