Task

On

NLP

**Course**: Artificial Intelligence

(Machine Learning & Deep Learning)

Name: Abbas Shafi

Father Name: Dr Muhammad Shafi

Week: 18

Submitted to: Sir Syed Nazir Afridi



National Vocational & Technical Training Commission

National Center For Big Data & Cloud Computing

University of Engineering & Technology

Peshawar

# 1 VocabMatching

[15]:

**import spacy**

[16]:

[17]:

[18]:

[19]:

*#solarpower* pattern1=[{'LOWER':'solarpower'}] *#solar power*

pattern2=[{'LOWER':'solar'},{'LOWER':'power'}]

*#solar-power*

pattern3=[{'LOWER':'solar'},{'IS\_PUNCT':**True**},{'LOWER':'power'}]

[20]:

[24]:

[23]:

[(8656102463236116519, 1, 3), (8656102463236116519, 11, 12),

(8656102463236116519, 14, 17)]

[23]: 'SolarPower'

[25]:

[26]:

8656102463236116519 SolarPower 1 3 Solar power

8656102463236116519 SolarPower 11 12 solarpower

8656102463236116519 SolarPower 14 17 Solar-Power

[ ]:

[27]:

[28]:

<IPython.core.display.HTML object> Saving phraseverb.txt to phraseverb.txt

[29]:

[31]:

Little Johnny hated going to see the dentist. It wasn't that his dentist was nasty; it was that Johnny wasn't too fond of sweets.

His dentist had warned him that his teeth would fall out if he kept up eating candy. Time and time again, the dentist had told him to cut out sweet food or at least cut down on the amount he ate.

As he lay down in the dentist's chair, all the horrible memories from his last visit came back to him. On that occasion, the dentist had to pull out one of his teeth! The pain was terrible - even with the anaesthetic the dentist had given him. When the anesthetic wore off it was difficult for him to eat or chew anything at all.

This time the check up was much better. His dentist checked out his teeth, made him wash out his mouth with pink liquid and then told him to spit it out into the sink. That was it. No problems and no pain! Johnny was delighted and so was his dentist. Johnny has finally learned his lesson and was taking better care of his teeth. Well done, Johnny!

[32]:

[34]:

[36]:

[37]:

[38]:

[ ]:

[(3680293220734633682, 39, 41), (3680293220734633682, 59, 61),

(3680293220734633682, 93, 95), (3680293220734633682, 106, 108),

(3680293220734633682, 150, 152)]

[ ]:

[ ]:

[ ]:

3680293220734633682 EconMatcher 39 41 fall out

3680293220734633682 EconMatcher 59 61 cut out

3680293220734633682 EconMatcher 93 95 came back

3680293220734633682 EconMatcher 106 108 pull out

3680293220734633682 EconMatcher 150 152 check up

# 2 Stemming

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

# 3 Lemmitization

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

sample="Once a fox was very hungry. He was roaming about in search of food. It␣

*‹→*reached a garden. Some branches of ripe grapes were hanging down a vine. The␣

*‹→*mouth of the fox watered to eat them.But the garden wall was very low. It␣

*‹→*jumped over the garden wall to reach the branches of grapes. He could not␣

*‹→*reach them. He jumped again and again but failed to get at them. He was␣

*‹→*completely tired. He realized that it was impossible for it to reach the␣

*‹→*grapes. He went away saying, they are sour and not good to eat. I eat them,␣

*‹→*I shall fall ill."

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

*#Lemmatization*

**for** i **in** range(len(sentences)): words=nltk.word\_tokenize(sentences[i]) *#applying the list comprehension*

words=[lemmatizer.lemmatize(word) **for** word **in** words **if** word **not in**␣

*‹→*set(stopwords.words('english'))]

sentences[i]=' '.join(words) print(sentences[i])

[ ]:

[ ]:

# 4 Bag of Words

[ ]:

[ ]:

*#Cleaning the text*

**import re**

**from nltk.stem.porter import** PorterStemmer

**from nltk.corpus import** stopwords

**from nltk.stem import** WordNetLemmatizer

ps=PorterStemmer() wordnet=WordNetLemmatizer() sentences=nltk.sent\_tokenize(para) corpus=[]

**for** i **in** range(len(sentences)): review=re.sub('[^**\a**-zA-Z]','', sentences[i]) review=review.lower()

review=review.split()

review=[ps.stem(word) **for** word **in** review **if** word **not in** set(stopwords.

*‹→*words('english'))]

review= ' '.join(review) corpus.append(review)

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: