

```
In [1]: import spacy
nlp = spacy.load('en_core_web_sm')
nlp
```

```
Out[1]: <spacy.lang.en.English at 0x1ef134eb490>
```

```
In [2]: #reading a basic text file
file_name = 'example.txt'
example_file_text = open(file_name).read()

#making an nlp object from the file
example_file_doc = nlp(example_file_text)
# Extract tokens for the given doc
print ([token.text for token in example_file_doc])
```

```
['Govind', 'grasped', 'tightly', 'to', 'the', 'top', 'of', 'the', 'seat', 'in',
'front', 'of', 'him', 'to', 'prevent', 'himself', 'from', 'being', 'jostled',
'off', 'his', 'place', ',', 'his', 'knuckles', 'white', 'due', 'to', 'his', 'ti
ght', 'grip', '.', 'The', 'bus', 'made', 'headway', 'with', 'the', 'driver', 't
rying', 'to', 'keep', 'it', 'steady', 'on', 'the', 'narrow', ',', 'partly', 'un
paved', 'road', '.', 'He', 'shook', 'his', 'head', ',', 'disgruntled', 'and',
'murmured', 'beneath', 'his', 'breath', ',', '"', 'Kuch', 'nahin', 'badla',
',', '"', '(', 'Nothing', 'has', 'changed', ')', '.', '\n\t', 'He', 'had', 'obs
tinately', 'left', 'the', 'village', 'to', 'taste', 'the', 'alleged', 'fast',
'and', 'easy', 'success', 'in', 'the', 'city', '.', 'He', 'had', 'promised', 't
o', 'return', 'within', 'months', '.', 'Today', ',', 'after', 'eight', 'years',
',', 'he', 'was', 'heading', 'home', 'without', 'having', 'made', 'even', 'a',
'slight', 'dent', 'in', 'his', 'fate', '.', '\n', 'He', 'had', 'done', 'it', 'a
ll-', 'cleaned', 'cars', ',', 'driven', 'trucks', ',', 'even', 'apprenticed',
'as', 'a', 'peon', ',', 'but', 'success', 'steered', 'clear', '.', 'The', 'deci
sion', 'to', 'return', 'had', 'been', 'demanding', '.', 'Govind', 'had', 'spen
t', 'many', 'a', 'sleepless', 'night', 'pondering', 'over', 'the', 'right', 'th
ing', 'to', 'do', 'under', 'the', 'woeful', 'circumstances', '.', 'A', 'millio
n', 'questions', 'raced', 'his', 'mind', '.', 'What', 'would', 'everyone', 'sa
y', '?', 'Did', 'he', 'not', 'feel', 'the', 'need', 'to', 'communicate', 'wit
h', 'them', '?', 'Why', 'was', 'he', 'back', 'now', '?', 'He', 'had', 'no', 'an
swers', 'then', ',', 'and', 'he', 'had', 'none', 'now', '.', '\n']
```

In [3]: *#print tokens and thier frequency by iterating on the Doc object*

```
for token in example_file_doc:
    print (token, token.idx)
```

```
Govind 0
grasped 7
tightly 15
to 23
the 26
top 30
of 34
the 37
seat 41
in 46
front 49
of 55
him 58
to 62
prevent 65
himself 73
from 81
being 86
jostled 92
of 100
```

In [4]: *#printing stop words in spacy core*

```
spacy_stopwords = spacy.lang.en.stop_words.STOP_WORDS
len(spacy_stopwords)
for stop_word in list(spacy_stopwords)[:12]:
    print(stop_word)

print ("-----")

for token in example_file_doc:
    if not token.is_stop:
        print(token)
```

```
whereupon
three
almost
itself
something
could
be
part
amongst
not
us
herself
-----
Govind
grasped
tightly
seat
prevent
jostled
```

```
In [5]: about_no_stopword_doc = [token for token in example_file_doc if not token.is_stop]
print (about_no_stopword_doc)
```

```
[Govind, grasped, tightly, seat, prevent, jostled, place, ,, knuckles, white, tight, grip, ., bus, headway, driver, trying, steady, narrow, ,, partly, unpaved, road, ., shook, head, ,, disgruntled, murmured, beneath, breath, ,, ", Kuch, nahin, badla, ., ", (, changed, ), .,
    , obstinately, left, village, taste, alleged, fast, easy, success, city, ., promised, return, months, ., Today, ,, years, ,, heading, home, having, slight, dent, fate, .,
    , all-, cleaned, cars, ,, driven, trucks, ,, apprenticed, peon, ,, success, steered, clear, ., decision, return, demanding, ., Govind, spent, sleepless, night, pondering, right, thing, woeful, circumstances, ., million, questions, raced, mind, ., ?, feel, need, communicate, ?, ?, answers, ,, .,
]
```

```
In [6]: #Lemmatization - reducing inflected forms of a word while still ensuring that the
#organizes, organized and organizing are all forms of organize
for token in about_no_stopword_doc:
    print (token, token.lemma_)
```

```
Govind Govind
grasped grasp
tightly tightly
seat seat
prevent prevent
jostled jostle
place place
, ,
knuckles knuckle
white white
tight tight
grip grip
. .
bus bus
headway headway
driver driver
trying try
steady steady
narrow narrow
```

```

In [7]: # taking the text file again and doing a word analysis

from collections import Counter

complete_text = example_file_text
complete_doc = nlp(complete_text)

# Remove stop words and punctuation symbols

words = [token.text for token in complete_doc if not token.is_stop and not token.is_punct]
word_freq = Counter(words)

# 5 commonly occurring words with their frequencies
common_words = word_freq.most_common(5)
print (common_words)
print("-----")

# Unique words
unique_words = [word for (word, freq) in word_freq.items() if freq == 1]
print (unique_words)

[('Govind', 2), ('success', 2), ('return', 2), ('\n', 2), ('grasped', 1)]
-----
['grasped', 'tightly', 'seat', 'prevent', 'jostled', 'place', 'knuckles', 'white', 'tight', 'grip', 'bus', 'headway', 'driver', 'trying', 'steady', 'narrow', 'partly', 'unpaved', 'road', 'shook', 'head', 'disgruntled', 'murmured', 'beneath', 'breath', 'Kuch', 'nahin', 'badla', 'changed', '\n\t', 'obstinately', 'left', 'village', 'taste', 'alleged', 'fast', 'easy', 'city', 'promised', 'months', 'Today', 'years', 'heading', 'home', 'having', 'slight', 'dent', 'fate', 'all-', 'cleaned', 'cars', 'driven', 'trucks', 'apprenticed', 'peon', 'steered', 'clear', 'decision', 'demanding', 'spent', 'sleepless', 'night', 'pondering', 'right', 'thing', 'woeful', 'circumstances', 'million', 'questions', 'raced', 'mind', 'feel', 'need', 'communicate', 'answers']

```

```
In [8]: from spacy import displacy
about_interest_text = example_file_text
about_interest_doc = nlp(about_interest_text)
displacy.serve(about_interest_doc, style='dep')

#displacy.render(about_interest_doc, style='dep', jupyter=True)
```

C:\ProgramData\Anaconda3\lib\site-packages\spacy\displacy__init__.py:94: UserWarning: [W011] It looks like you're calling displacy.serve from within a Jupyter notebook or a similar environment. This likely means you're already running a local web server, so there's no need to make displacy start another one. Instead, you should be able to replace displacy.serve with displacy.render to show the visualization.

warnings.warn(Warnings.W011)

Govind PROPEN grasped VERB tightly ADV to ADP the DET top NOUN of ADP the DET seat NOUN in ADP front NOUN of ADP him PRON to PART prevent VERB himself PRON from ADP being AUX jostled VERB off ADP his DET place, NOUN his DET knuckles NOUN white ADJ due ADP to ADP his DET tight ADJ grip. NOUN The DET bus NOUN made VERB headway NOUN with ADP the DET driver NOUN trying VERB to PART keep VERB it PRON steady ADJ on ADP the DET narrow, ADJ partly ADV unpaved ADJ road. NOUN He PRON shook VERB his DET head, NOUN disgruntled VERB and CCONJ murmured VERB beneath ADP his DET breath, " NOUN Kuch PROPEN nahin ADJ badla." (NOUN Nothing PRON has AUX changed). VERB SPACE He PRON had AUX obstinately ADV left VERB the DET village NOUN to PART taste VERB the DET alleged VERB fast ADJ and CCONJ easy ADJ success NOUN in ADP the DET city. NOUN He PRON had AUX promised VERB to PART return VERB within ADP months. NOUN Today, NOUN after ADP eight NUM years, NOUN he PRON was AUX heading VERB home ADV without ADP having VERB made VERB even ADV a DET slight ADJ dent NOUN in ADP his DET fate. NOUN SPACE He PRON had AUX done VERB it PRON all- ADV cleaned VERB cars, NOUN driven VERB trucks, NOUN even ADV apprenticed VERB as SCONJ a DET peon, NOUN but CCONJ success NOUN steered VERB clear. ADV The DET decision NOUN to PART return VERB had AUX been AUX demanding. VERB Govind PROPEN had AUX spent VERB many DET a DET sleepless NOUN night NOUN pondering VERB over ADP the DET right ADJ thing NOUN to PART do AUX under ADP the DET woeful ADJ circumstances. NOUN A DET million NUM questions NOUN raced VERB his DET mind. NOUN What PRON would VERB everyone PRON say? VERB Did AUX he PRON not PART feel VERB the DET need NOUN to PART communicate VERB with ADP them? PRON Why ADV was AUX he PRON back ADV now? ADV He PRON had AUX no DET answers NOUN then, ADV and CCONJ he PRON had AUX none NOUN now. ADV SPACE nsubj advmod prep det pobj prep det pobj prep pobj prep pobj aux advcl dobj prep auxpass pcomp prep poss pobj poss nsubj conj prep pcomp poss amod pobj det nsubj dobj prep det pobj acl aux xcomp dobj oprd prep det amod advmod amod pobj nsubj poss dobj advcl cc conj prep poss pobj compound compound dep nsubj aux nsubj aux advmod det dobj aux xcomp det amod amod cc conj dobj prep det pobj nsubj aux aux xcomp prep pobj npadvmod prep nummod pobj nsubj aux advmod prep aux pcomp advmod det amod dobj prep poss punct nsubj aux dobj det amod dobj amod appos advmod acl prep det pobj cc nsubj conj oprd det nsubj aux acl aux aux nsubj aux amod det compound dobj xcomp prep det amod pobj aux advcl prep det amod pobj quantmod nummod nsubj poss dobj dobj aux nsubj aux nsubj neg det dobj aux acl prep pobj advmod nsubj advmod advmod nsubj det dobj advmod cc nsubj conj dobj advmod

Using the 'dep' visualizer

Serving on <http://0.0.0.0:5000> (<http://0.0.0.0:5000>) ...

Shutting down server on port 5000.

In [26]: *#Named Entity Recognition (NER) is the process of locating named entities in unstructured text and classifying them into pre-defined categories, such as person names, organizations, locations, and time expressions, etc*

```
home_bound_text = ("Govind grasped tightly to the top of the seat in front of him")
home_bound_doc = nlp(home_bound_text)
for ent in home_bound_doc.ents:
    print(ent.text, ent.start_char, ent.end_char, ent.label_, spacy.explain(ent.label_))
```

Govind 0 6 PERSON People, including fictional

In []: