**TOKENIZER – Automatically extract the sentence start indexes of tokens**

**SPACY**: is a free and open-source library for \*\*Natural Language Processing\*\* (NLP) in Python with a lot of in-built capabilities. It’s becoming increasingly popular for processing and analyzing data in NLP. Unstructured textual data is produced at a large scale, and it’s important to process and derive insights from unstructured data. To do that, you need to represent the data in a format that can be understood by computers. NLP can help you do that.

***How to Install spaCy***

spaCy can be installed using pip, a Python package manager. You can use a virtual environment to avoid depending on system-wide packages.

Create a new virtual environment in a folder:

1. Create a new folder in desktop and navigate to that directory in CMD.

cd C:\Users\XYZ\Desktop\tools and then in CMD type:

*If Python****virtualenv****is not already installed, execute the following code in the console:*

pip install virtualenv

Then create a new virtual env

python3 -m venv venv

1. Activate this virtual environment and install spaCy:

cd venv

.\Scripts\activate

pip install spacy

How to Download Models and Data

English: python -m spacy download en\_core\_web\_sm

German: python -m spacy download de\_core\_news\_sm

Now in CMD, type python:

Python Commands: >>> import spacy

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

### How to Read a Text File and generate Tokens:

file\_name = 'introduction.txt'

introduction\_file\_text = open(file\_name).read()

introduction\_file\_doc = nlp(introduction\_file\_text)

# Extract tokens for the given doc

print ([token.text for token in introduction\_file\_doc])

# Extract token and sentence start for the given doc

>>> for token in about\_doc:

... print (token, token.idx)

Press Enter….

Output: Gus 0

Proto 4

is 10

a 13

Python 15

developer 22

currently 32

working 42

for 50

## **Word Frequency**

You can now convert a given text into tokens and perform statistical analysis over it. This analysis can give you various insights about word patterns, such as common words or unique words in the 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)

[('Gus', 4), ('London', 3), ('Natural', 3), ('Language', 3), ('Processing', 3)]

>>> # Unique words

>>> unique\_words = [word for (word, freq) in word\_freq.items() if freq == 1]

>>> print (unique\_words)

['Proto', 'currently', 'working', 'based', 'company',

'interested', 'conference', 'happening', '21', 'July',

'2019', 'titled', 'Applications', 'helpline', 'number',

'available', '+1', '1234567891', 'helping', 'organize',

'keeps', 'organizing', 'local', 'meetups', 'internal',

'talks', 'workplace', 'presenting', 'introduce', 'reader',

'Use', 'cases', 'Apart', 'work', 'passionate', 'music', 'play',

'enrolled', 'weekend', 'batch', 'situated', 'Mayfair', 'City']

>>> words\_all = [token.text for token in complete\_doc if not token.is\_punct]

>>> word\_freq\_all = Counter(words\_all)

>>> # 5 commonly occurring words with their frequencies

>>> common\_words\_all = word\_freq\_all.most\_common(5)

>>> print (common\_words\_all)

[('is', 10), ('a', 5), ('in', 5), ('Gus', 4), ('of', 4)]

## **Part of Speech Tagging**

>>> for token in about\_doc:

... print (token, token.tag\_, token.pos\_, spacy.explain(token.tag\_))

>>> nouns = []

>>> adjectives = []

>>> for token in about\_doc:

... if token.pos\_ == 'NOUN':

... nouns.append(token)

... if token.pos\_ == 'ADJ':

... adjectives.append(token)

...

>>> nouns

[developer, company]

>>> adjectives

[interested]

## **Named Entity Recognition**

>>> piano\_class\_text = ('Great Piano Academy is situated'

... ' in Mayfair or the City of London and has'

... ' world-class piano instructors.')

>>> piano\_class\_doc = nlp(piano\_class\_text)

>>> for ent in piano\_class\_doc.ents:

... print(ent.text, ent.start\_char, ent.end\_char,

... ent.label\_, spacy.explain(ent.label\_))

...

Great Piano Academy 0 19 ORG Companies, agencies, institutions, etc.

Mayfair 35 42 GPE Countries, cities, states

the City of London 46 64 GPE Countries, cities, states

Redact people’s names from a text

>>> survey\_text = ('Out of 5 people surveyed, James Robert,'

... ' Julie Fuller and Benjamin Brooks like'

... ' apples. Kelly Cox and Matthew Evans'

... ' like oranges.')

...

>>> def replace\_person\_names(token):

... if token.ent\_iob != 0 and token.ent\_type\_ == 'PERSON':

... return '[REDACTED] '

... return token.string

...

>>> def redact\_names(nlp\_doc):

... for ent in nlp\_doc.ents:

... ent.merge()

... tokens = map(replace\_person\_names, nlp\_doc)

... return ''.join(tokens)

...

>>> survey\_doc = nlp(survey\_text)

>>> redact\_names(survey\_doc)

'Out of 5 people surveyed, [REDACTED] , [REDACTED] and'

' [REDACTED] like apples. [REDACTED] and [REDACTED]'

' like oranges.'

2 files are included: tsv2json and txt2Json for automatic conversion of TSV and TXT files to JSON format for easy CATMA Import.

To start the tokenizer, please change the format of the output file which needs to be converted with the two headers: token and sentstart.

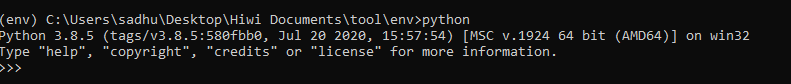
**How to run tokenizer?**

Navigate to the virtual env created above and activate it using the .bat file



**Type ‘Python’ without quotes:**

**You should see something like this:**



**Paste the following code from *tokenize.py***

import spacy

import json

import sys

nlp = spacy.load('de\_core\_news\_sm')

file\_name = 'sandmann.txt'

file\_text = open(file\_name).read()

file\_doc = nlp(file\_text)

for token in file\_doc:

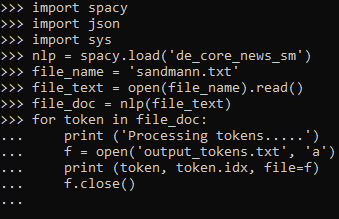
print ('Processing tokens.....')

f = open('output\_tokens.txt', 'a')

print (token, token.idx, file=f)

f.close()

**Press enter twice then you should see…**



**The token file should be created inside env folder.**

**How to convert tokens output to JSON?**

Run the txt2Json.py file from cmd.

**Navigate to your root folder:**

**Please make sure the file which you want to convert must have the header at the beginning of the file:**

**token sentstart**

Der 0

Sandmann 4

Nathanael 13

an 23 and so on.....

**Then type in CMD:** python txt2Json.py output.txt *where* ***output.txt*** *is the filename which contains the tokenized sentence start indexes in plain text format.*



The Output\_tokens file should be created now.