Natural Language Processing

This section some tools to process and work with text in Python.

TextBlob: Processing Text in One Line of Code

Processing text doesn't need to be hard. If you want to find the sentiment of the text, tokenize text, find noun phrase and word frequencies, correct spelling, etc in one line of code, try TextBlob.

```
!pip install textblob
```

```
!python -m textblob.download_corpora
```

```
>>> from textblob import TextBlob

>>> text = "Today is a beautiful day"
>>> blob = TextBlob(text)

>>> blob.words # Word tokenization
```

```
WordList(['Today', 'is', 'a', 'beautiful', 'day'])
```

```
>>> blob.noun_phrases # Noun phrase extraction
```

```
WordList(['beautiful day'])
```

```
>>> blob.sentiment # Sentiment analysis
```

```
Sentiment(polarity=0.85, subjectivity=1.0)
```

```
>>> blob.word_counts # Word counts
```

```
defaultdict(int, {'today': 1, 'is': 1, 'a': 1, 'beautiful': 1,
'day': 1})
```

```
# Spelling correction
>>> text = "Today is a beutiful day"
>>> blob = TextBlob(text)
>>> blob.correct()
```

```
TextBlob("Today is a beautiful day")
```

Link to TextBlob.

Link to my article about TextBlob.

sumy: Summarize Text in One Line of Code

```
!pip install sumy
```

If you want to summarize text using Python or command line, try sumy.

Below is how sumy summarizes the article How to Learn Data Science (Step-By-Step) in 2020 at DataQuest.

```
$ sumy lex-rank --length=10 --
url=https://www.dataquest.io/blog/learn-data-science/
```

```
So how do you start to learn data science?
If I had started learning data science this way, I never would
have kept going.
I learn when Iâ€™m motivated, and when I know why Iâ€™m
learning something.
Thereâ€™s some science behind this, too.
If you want to learn data science or just pick up some data
science skills, your first goal should be to learn to love
data.
But itâ€™s important to find that thing that makes you want to
learn.
By working on projects, you gain skills that are immediately
applicable and useful, because real-world data scientists have
to see data science projects through from start to finish, and
most of that work is in fundamentals like cleaning and
managing the data.
And so on, until the algorithm worked well.
Find people to work with at meetups.
For more information on these, you can take a look at our Data
Scientist learning path , which is designed to teach all of
the important data science skills for Python learners.
```

Link to Sumy.

Spacy_streamlit: Create a Web App to Visualize Your Text in 3 Lines of Code

```
!pip install spacy-streamlit
```

If you want to quickly create an app to visualize the structure of a text, try spacy_streamlit.

To understand how to use spacy_streamlit, we add the code below to a file called `streamlit_app.py`:

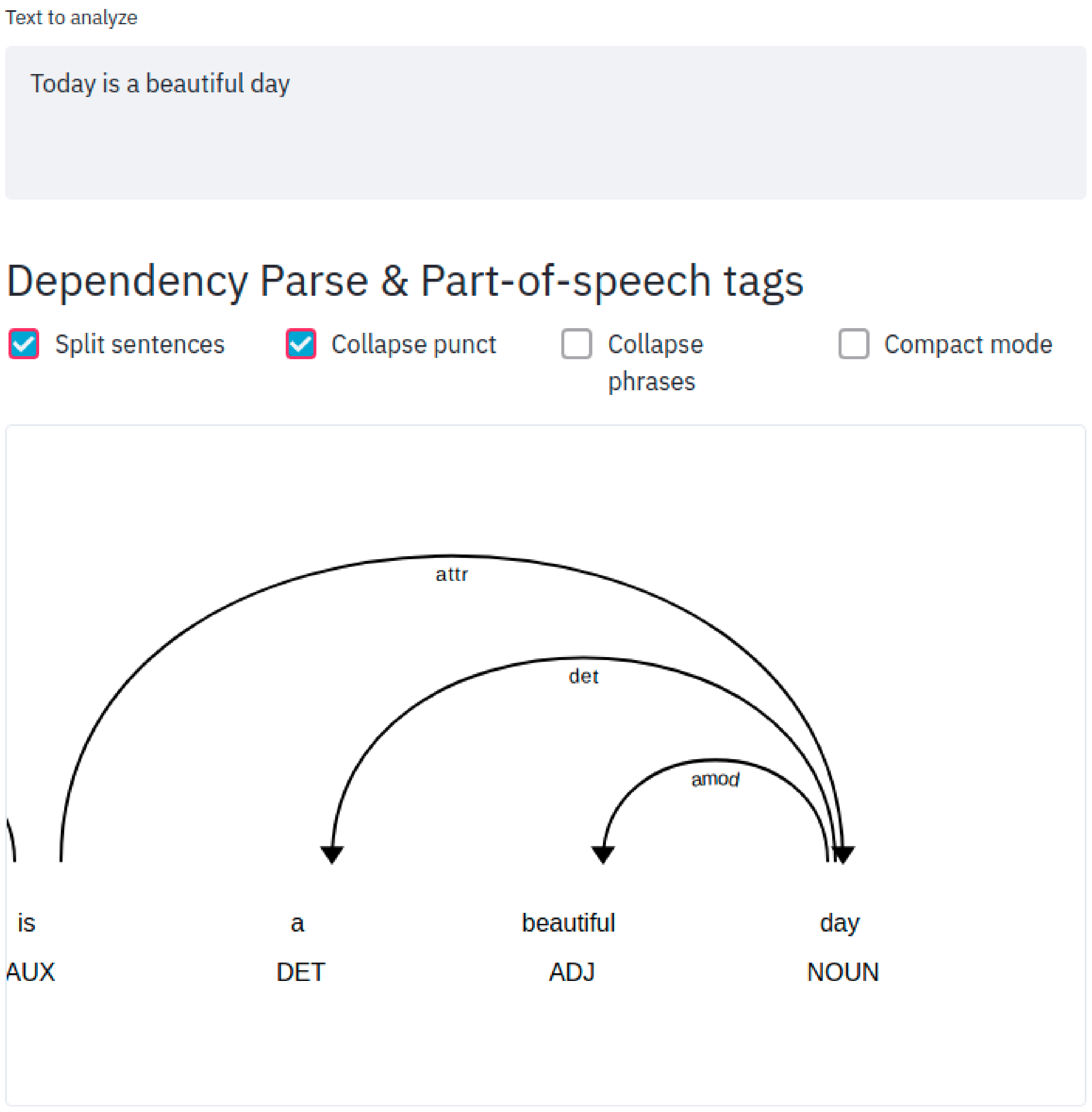
```
# streamlit_app.py
import spacy_streamlit

models = ['en_core_web_sm']
text = "Today is a beautiful day"
spacy_streamlit.visualize(models, text)
```

On your terminal, type:

```
$ streamlit run streamlit_app.py
```

Click the URL generated by spacy_streamlit and you should see something like below:



Text to analyze

Today is a beautiful day

Dependency Parse & Part-of-speech tags

[✓ Split sentences] [✓ Collapse punct] Collapse phrases Compact mode

Link to spacy-streamlit.

textacy: Extract a Contiguous Sequence of 2 Words

```
!pip install spacy textacy
```

If you want to extract a contiguous sequence of 2 words, for example, 'data science', not 'data', what should you do? That is when the concept of extracting n-gram from text becomes useful.

A really useful tool to easily extract n-gram with a specified number of words in the sequence is `textacy`.

```
import pandas as pd
import spacy
from textacy.extract import ngrams

nlp = spacy.load('en_core_web_sm')

text = nlp('Data science is an inter-disciplinary field that
uses scientific methods, processes, algorithms, and systme to
extract knowledge and insights from many structural and
unstructured data.')

n_grams = 2 # contiguous sequence of a word
min_freq = 1 # extract n -gram based on its frequency

pd.Series([n.text for n in ngrams(text, n=n_grams,
min_freq=1)]).value_counts()
```

```
Data science          1
disciplinary field    1
uses scientific       1
scientific methods    1
extract knowledge     1
unstructured data     1
dtype: int64
```

Link to textacy

Convert Number to Words

```
!pip install num2words
```

If there are both number 105 and the words 'one hundred and five' in a text, they should deliver the same meaning. How can we map 105 to 'one hundred and five'? There is a Python libary to convert number to words called `num2words`.

```
>>> from num2words import num2words

>>> num2words(105)
```

```
'one hundred and five'
```

```
>>> num2words(105, to='ordinal')
```

```
'one hundred and fifth'
```

The library can also generate ordinal numbers and support multiple languages!

```
>>> num2words(105, lang='vi')
```

```
một trăm lẻ năm
```

```
>>> num2words(105, lang='es')
```

```
'ciento cinco'
```

Link to num2words.

texthero.clean: Preprocess Text in One Line of Code

```
!pip install texthero
```

If you want to preprocess text in one line of code, try texthero. The `texthero.clean` method will:

- fill missing values
- convert upper case to lower case
- remove digits
- remove punctuation
- remove stopwords
- remove whitespace

The code below shows an example of `texthero.clean`.

```
import numpy as np
import pandas as pd
import texthero as hero

df = pd.DataFrame(
    {
        "text": [
            "Today is a    beautiful day",
            "There are 3 ducks in this pond",
            "This is. very cool.",
            np.nan,
        ]
    }
)

df.text.pipe(hero.clean)
```

```
0    today beautiful day
1             ducks pond
2                   cool
3
Name: text, dtype: object
```

Texthero also provides other useful methods to process and visualize text.

Link to texthero.

wordfreq: Estimate the Frequency of a Word in 36 Languages

```
!pip install wordfreq
```

If you want to look up the frequency of a certain word in your language, try wordfreq.

wordfreq supports 36 languages. wordfreq even covers words that appear at least once per 10 million words.

```
>>> import matplotlib.pyplot as plt
>>> import seaborn as sns
>>> from wordfreq import word_frequency

>>> word_frequency("eat", "en")
```

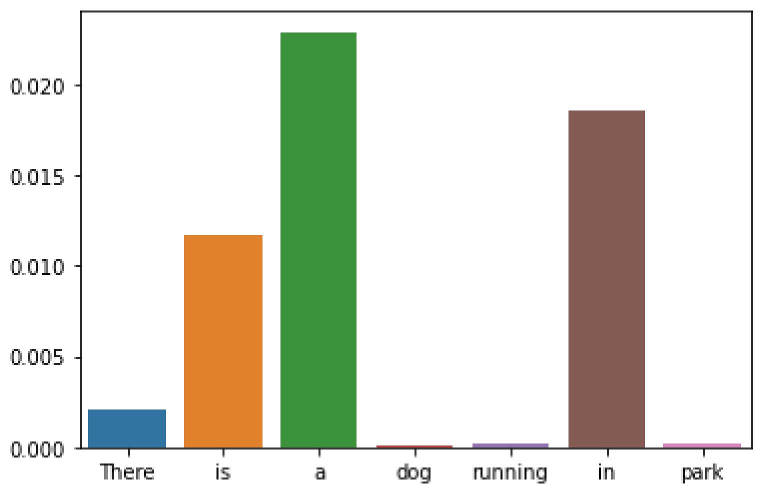
```
0.000135
```

```
>>> word_frequency("the", "en")
```

```
0.0537
```

```
>>> sentence = "There is a dog running in a park"
>>> words = sentence.split(" ")
>>> word_frequencies = [word_frequency(word, "en") for word in
words]

>>> sns.barplot(words, word_frequencies)
>>> plt.show()
```



Link to wordfreq.

newspaper3k: Extract Meaningful Information From an Articles in 2 Lines of Code

```
!pip install newspaper3k nltk
```

If you want to quickly extract meaningful information from an article in a few lines of code, try newspaper3k.

```
>>> from newspaper import Article
>>> import nltk

>>> nltk.download("punkt")
```

```
>>> url = "https://www.dataquest.io/blog/learn-data-science/"
>>> article = Article(url)
>>> article.download()
>>> article.parse()
```

```
>>> article.title
```

```
'How to Learn Data Science (A step-by-step guide)'
```

```
>>> article.publish_date
```

```
datetime.datetime(2020, 5, 4, 7, 1, tzinfo=tzutc())
```

```
>>> article.top_image
```

```
'https://www.dataquest.io/wp-content/uploads/2020/05/learn-
data-science.jpg'
```

```
>>> article.nlp()
```

```
>>> article.summary
```

```
'How to Learn Data ScienceSo how do you start to learn data
science?\nIf you want to learn data science or just pick up
some data science skills, your first goal should be to learn
to love data.\nRather, consider it as a rough set of
guidelines to follow as you learn data science on your own
path.\nI personally believe that anyone can learn data science
if they approach it with the right frame of mind.\nIâ€™m also
the founder of Dataquest, a site that helps you learn data
science in your browser.'
```

```
>>> article.keywords
```

```
['scientists',
 'guide',
 'learning',
 'youre',
 'science',
 'work',
 'skills',
 'youll',
 'data',
 'learn',
 'stepbystep',
 'need']
```

Link to newspaper3k.

Questgen.ai: Question Generator in Python

It can be time-consuming to generate questions for a document. Wouldn't it be nice if you can automatically generate questions using Python? That is when `Questgen.ai` comes in handy.

With a few lines of code, the questions for your document are automatically generated.

```python
from pprint import pprint
import nltk
nltk.download('stopwords')
from Questgen import main
```

```python
payload = {
    "input_text": """The weather today was nice so I went for
a walk. I stopped for a quick chat with my neighbor.
    It turned out that my neighbor just got a dog named
Pepper. It is a black Labrador Retriever."""
}
```

With `Questgen.ai`, you can either generate boolean questions:

```python
qe = main.BoolQGen()
output = qe.predict_boolq(payload)
pprint(output)
```

```
{'Boolean Questions': ['Is there a dog in my neighborhood?',
                       "Is pepper my neighbor's dog?",
                       'Is pepper the same as a labrador?'],
 'Count': 4,
 'Text': 'The weather today was nice so I went for a walk. I
stopped for a '
         'quick chat with my neighbor.\n'
         '    It turned out that my neighbor just got a dog
named Pepper. It '
         'is a black Labrador Retriever.'}
```

... or generate FAQ questions:

```
output = qg.predict_shortq(payload)
pprint(output)
```

```
Running model for generation
{'questions': [{'Question': 'What was the purpose of the
stop?', 'Answer': 'chat', 'id': 1, 'context': 'I stopped for a
quick chat with my neighbor.'}, {'Question': 'Who got a dog
named Pepper?', 'Answer': 'neighbor', 'id': 2, 'context': 'It
turned out that my neighbor just got a dog named Pepper. I
stopped for a quick chat with my neighbor.'}]}
{'questions': [{'Answer': 'chat',
                'Question': 'What was the purpose of the
stop?',
                'context': 'I stopped for a quick chat with my
neighbor.',
                'id': 1},
               {'Answer': 'neighbor',
                'Question': 'Who got a dog named Pepper?',
                'context': 'It turned out that my neighbor
just got a dog '
                           'named Pepper. I stopped for a
quick chat with my '
                           'neighbor.',
                'id': 2}],
 'statement': 'The weather today was nice so I went for a
walk. I stopped for '
              'a quick chat with my neighbor. It turned out
that my neighbor '
              'just got a dog named Pepper. It is a black
Labrador Retriever.'}
```

Link to Questgen.ai.