Spcay in NLP

Eng. Fatma

What Is spaCy?

- SpaCy is a free, open-source library for advanced NLP written in Python programming languages.
- SpaCy is designed specifically for production use and build NLP applications to process large volumes of text different from NLTK focused on teaching and learning perspective.

What Is spaCy?

- NLP-based statistical models for over 19 commonly used languages,
- Tokenization tools implementation for over 60 international languages,
- NLP pipeline components include NER, POS Tagging, DP, Text Classification,
- and Chatbot implementation,
- integration with common Python platforms such as TensorFlow, PyTorch and
- other high-level frameworks,
- integration with the latest Transformer and BERT technologies,

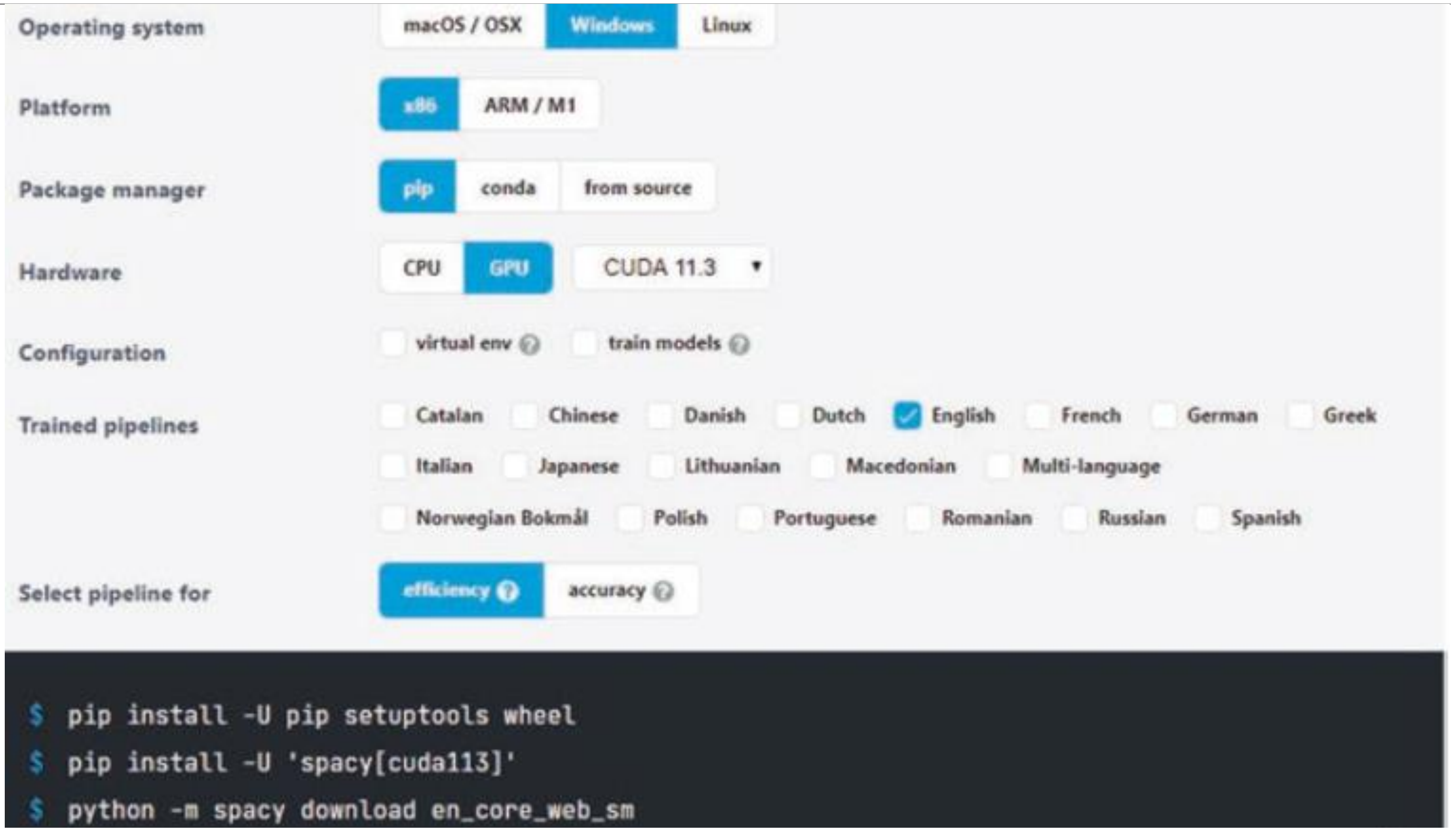
How to Install spaCy?

- Spacy official webste:

https://spacy.io/usage

- Screenshot of spaCy

configuration selection



Operating system: macOS / OSX [✓ Windows] Linux

Platform: [✓ x86] ARM / M1

Package manager: [✓ pip] conda from source

Hardware: CPU [✓ GPU] CUDA 11.3

Configuration: ☐ virtual env ☐ train models

Trained pipelines: ☐ Catalan ☐ Chinese ☐ Danish ☐ Dutch ✓ English ☐ French ☐ German ☐ Greek ☐ Italian ☐ Japanese ☐ Lithuanian ☐ Macedonian ☐ Multi-language ☐ Norwegian Bokmål ☐ Polish ☐ Portuguese ☐ Romanian ☐ Russian ☐ Spanish

Select pipeline for: [✓ efficiency] accuracy

```
$ pip install -U pip setuptools wheel
$ pip install -U 'spacy[cuda113]'
$ python -m spacy download en_core_web_sm
```

Tokenization using spaCy

Step 1: Import spaCy Module

```
import spacy
```

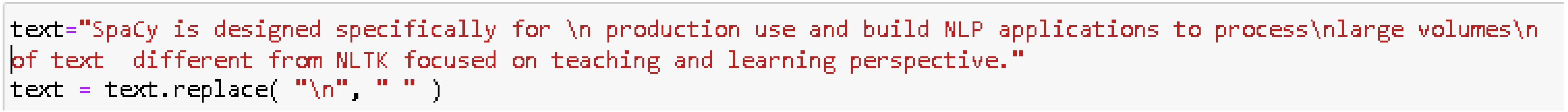
Step 2: Load spaCy Module "en_core_web_sm"
Use en_core_web_md-3.2.0 package for English pipeline optimized for CPU in the current platform with components including: tok2vec, tagger, parser, senter, ner, attribute_ruler, lemmatizer.

```
nlp = spacy.load( "en_core_web_sm" )
```

Tokenization using spaCy

Step3: Replace All Newline Symbols

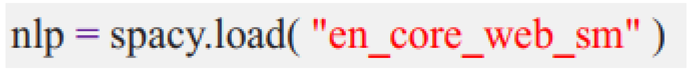
Note: Since text file already exists, skip try-except module to save programming steps



```
text="SpaCy is designed specifically for \n production use and build NLP applications to process\nlarge volumes\n
of text  different from NLTK focused on teaching and learning perspective."
text = text.replace( "\n", " " )
```

Step 4: Load spaCy Module "en_core_web_sm"

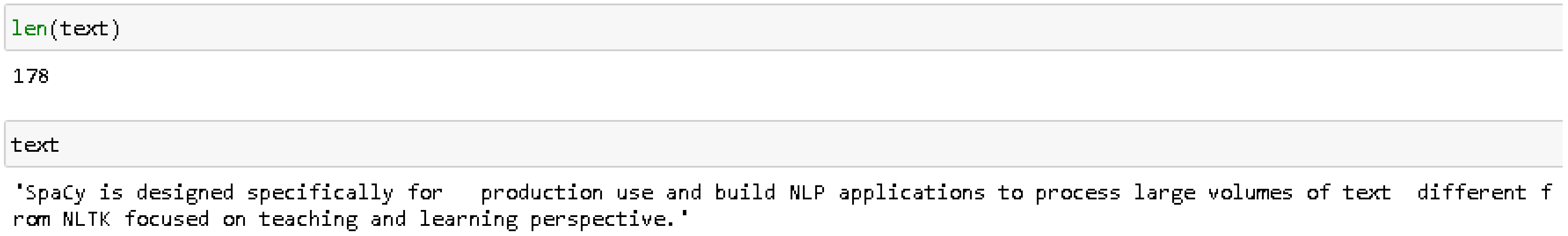
Use en_core_web_md-3.2.0 package for English pipeline optimized for CPU in the current platform with components including: tok2vec, tagger, parser, senter, ner, attribute_ruler, lemmatizer.



```
nlp = spacy.load( "en_core_web_sm" )
```

Tokenization using spaCy

Step5: Simple Counting



```
len(text)
```

```
178
```

```
text
```

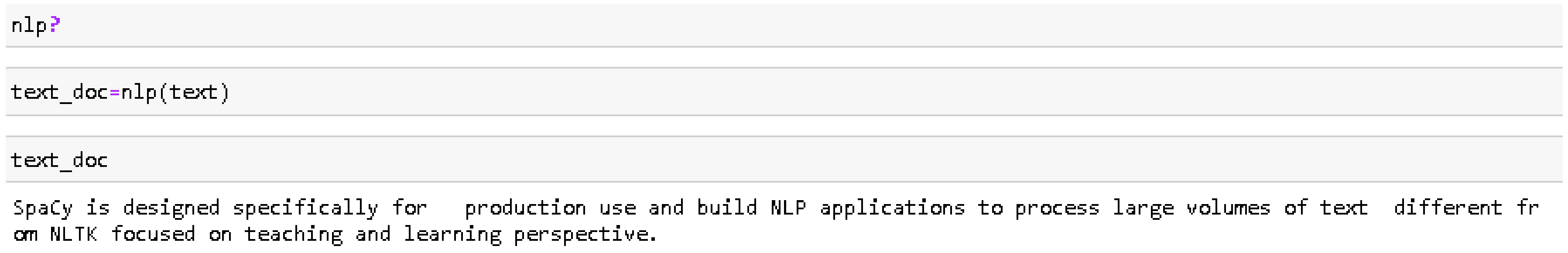
```
'SpaCy is designed specifically for   production use and build NLP applications to process large volumes of text  different f
rom NLTK focused on teaching and learning perspective.'
```

Tokenization using spaCy

Step 6: invoke nlp() Method in spaCy

SpaCy nlp() method is an important Text Processing Pipeline to initialize nlp object (English in our case) for NLP processing such as tokenization. It will convert any text string object into a nlp object.

Exercise: nlp() docstring , how it works.



```
nlp?
```

```
text_doc=nlp(text)
```

```
text_doc
```

```
SpaCy is designed specifically for   production use and build NLP applications to process large volumes of text  different fr
om NLTK focused on teaching and learning perspective.
```

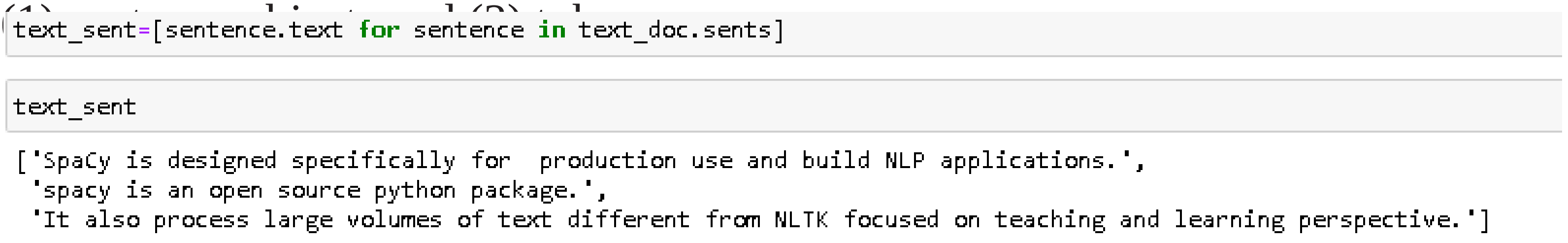
Tokenization using spaCy

Step 7: Convert Text Document Into Sentence Object

SpaCy is practical for text document tokenization to convert text document object

into

.



```
text_sent=[sentence.text for sentence in text_doc.sents]
```

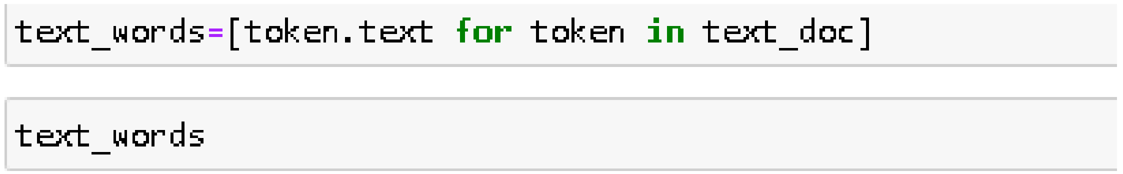
```
text_sent
```

```
['SpaCy is designed specifically for  production use and build NLP applications.',
 'spacy is an open source python package.',
 'It also process large volumes of text different from NLTK focused on teaching and learning perspective.']
```

Tokenization using spaCy

Step 8: Directly Tokenize Text Document

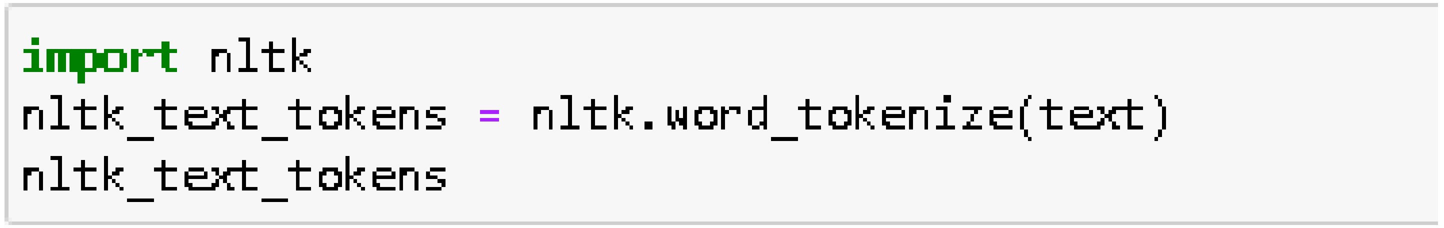
Tokenize text document into word tokens by using “token” object in spaCy instead of text document object extraction into sentence list object. Study how it operates.



```
text_words=[token.text for token in text_doc]
```

```
text_words
```

Nltk tokenization vs spacy tokenization



```
import nltk
nltk_text_tokens = nltk.word_tokenize(text)
nltk_text_tokens
```

spaCy tends to be faster than NLTK
if speed and efficiency are crucial factors for your tokenization tasks, spaCy is likely the better choice

Which one is faster!?

Exercise

Given an input text, removing stop words from a text using NLTK library.