Activity

- Write a function to find the frequency of the top X words in of the Reuters corpus.
- Report how many of these are stop words for the top 10, 20, and 30 words.

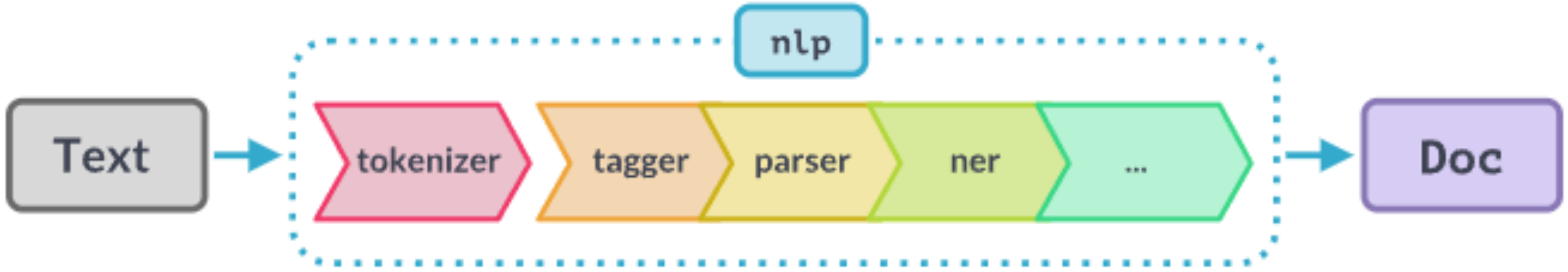
Topic 1
The art of preprocessing

spaCy

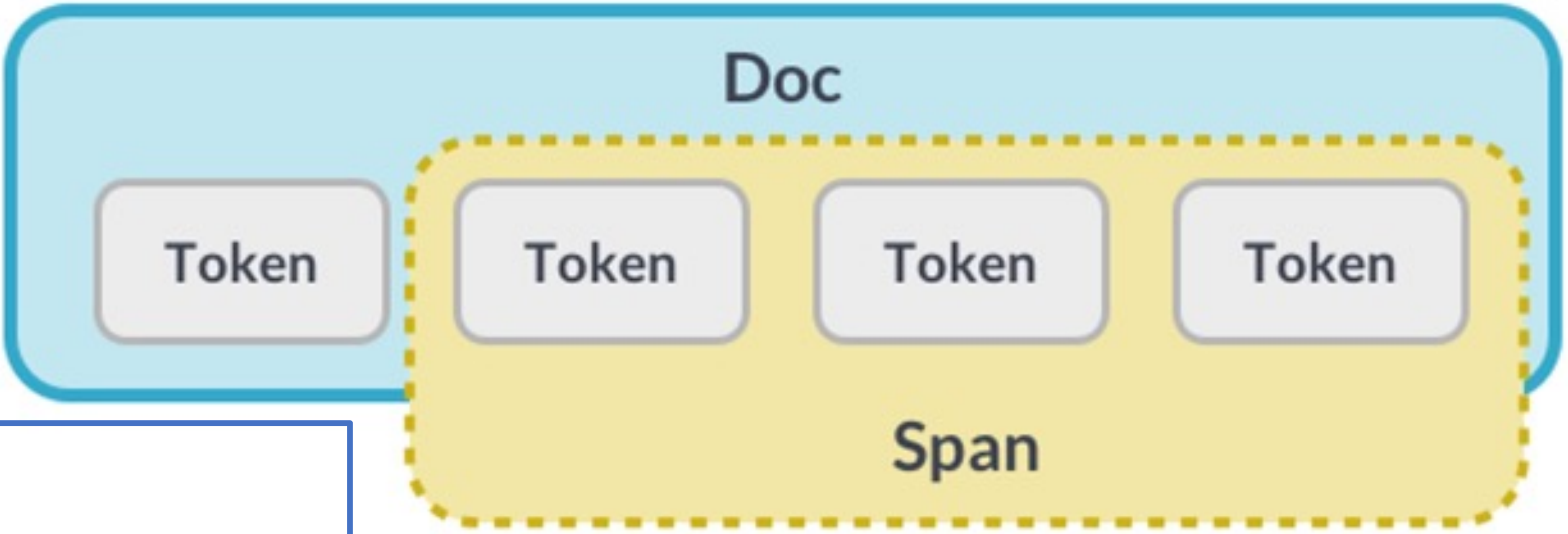
SpaCy

- SpaCy is an industrial-strength natural language processing (NLP) library for Python
- Written in Cython
- “Concise and user-friendly API”
- Documentation

SpaCy is object-oriented



Selecting tokens



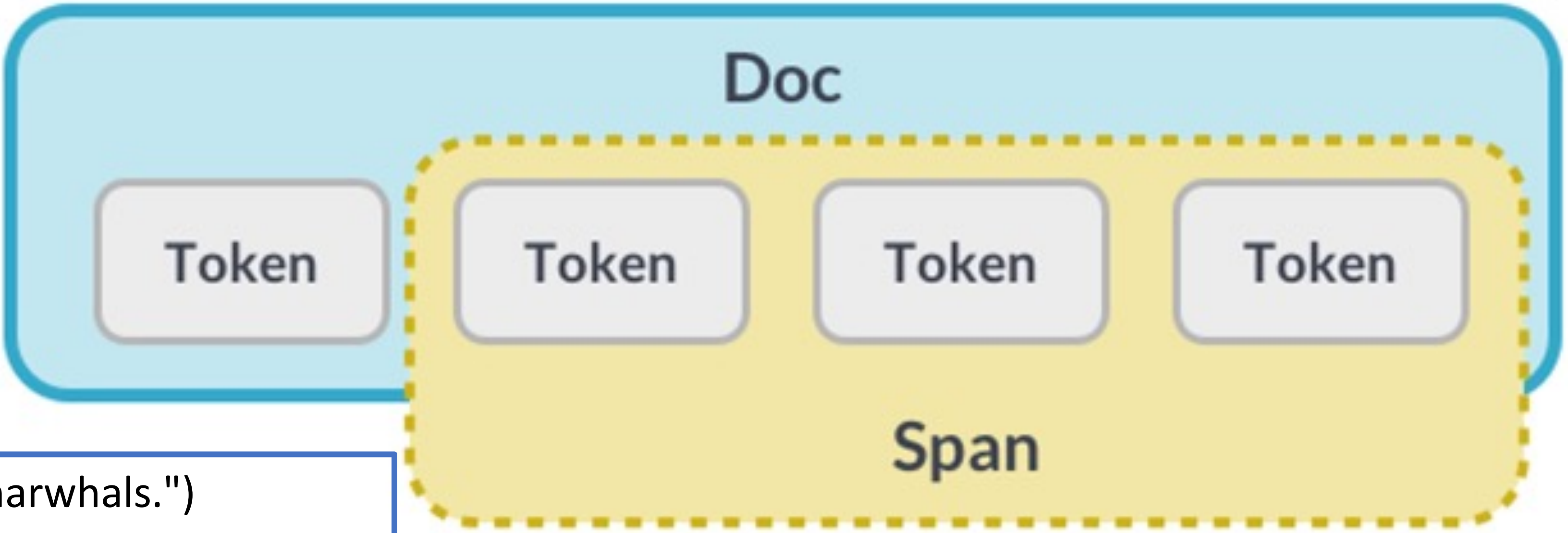
```
nlp = spacy.blank("en")

# processing the text
doc = nlp("I like tree kangaroos and narwhals.")

# the first token
first_token = doc[0]

# print the first token's text
print(first_token.text)
```

Spans



```
doc = nlp("I like tree kangaroos and narwhals.")

# slice of the Doc for "tree kangaroos"
tree_kangaroos = doc[2:4]

# slice of the Doc for "tree kangaroos and narwhals"
(without the ".")
tree_kangaroos_and_narwhals = doc[2:6]
```

Iterate over the tokens in the Doc

```
for token in doc:
    # does the token resembles a number?
    if token.like_num:
        # get the next token in the document
        next_token = doc[token.i + 1]
        # does the next token's text equal "%”?
        if next_token.text == "%":
            print("Percentage found:", token.text)
```

What are trained pipelines?

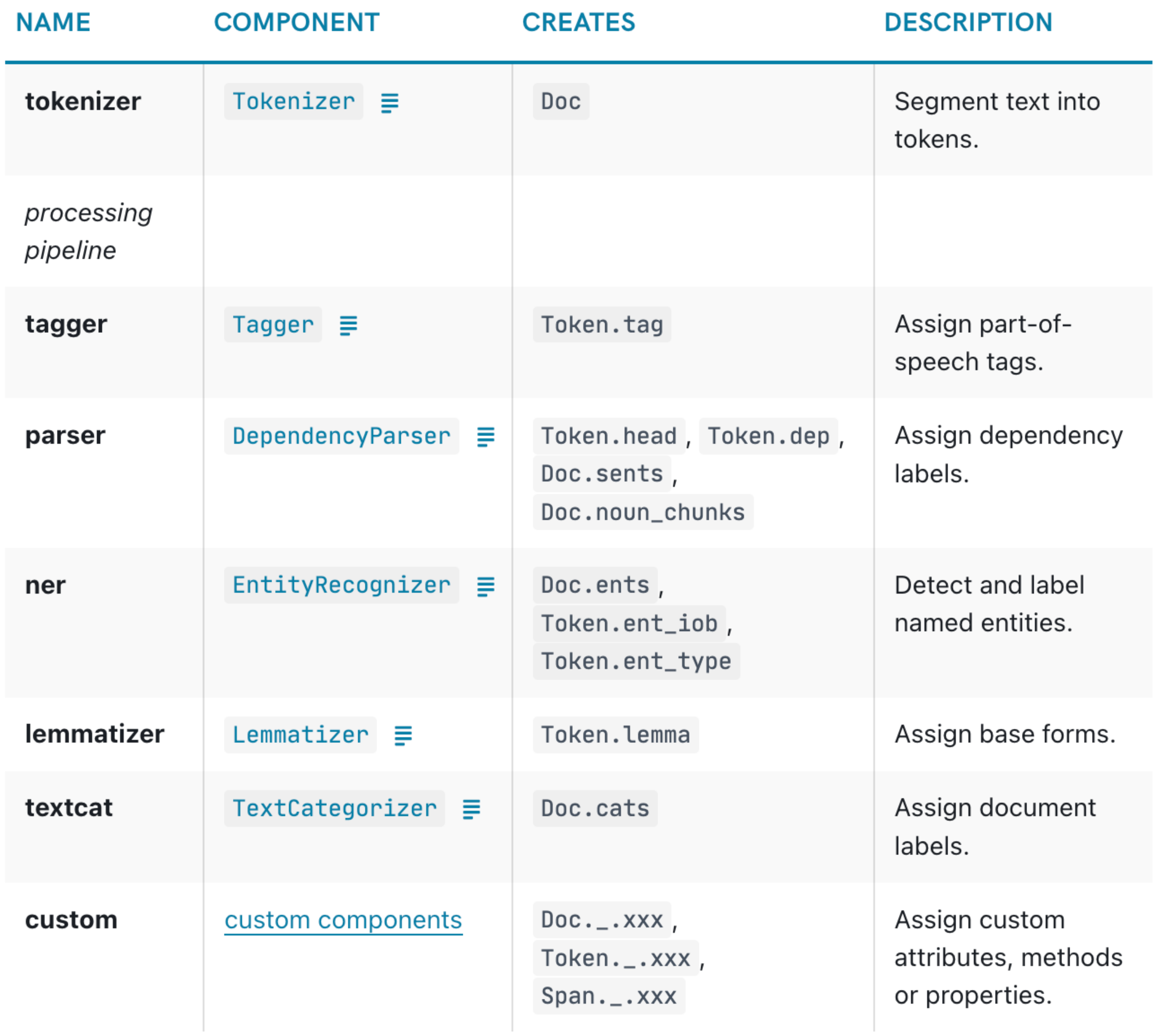
- Models that enable SpaCy to predict linguistic attributes in context
  - Part-of-speech tags
  - Syntactic dependencies
  - Named entities
- Trained on labeled example texts
- Can be updated with more examples to fine-tune predictions
- e.g., spacy.load('en_core_web_sm’)

You don’t need to include all the bells and whistles if you don’t want to

- Lets say we just want to tokenize the words
- Make your own custom pipeline
- spacy.load('en_core_web_sm’) vs spacy.load(‘blank’)

Ordering of pipeline components

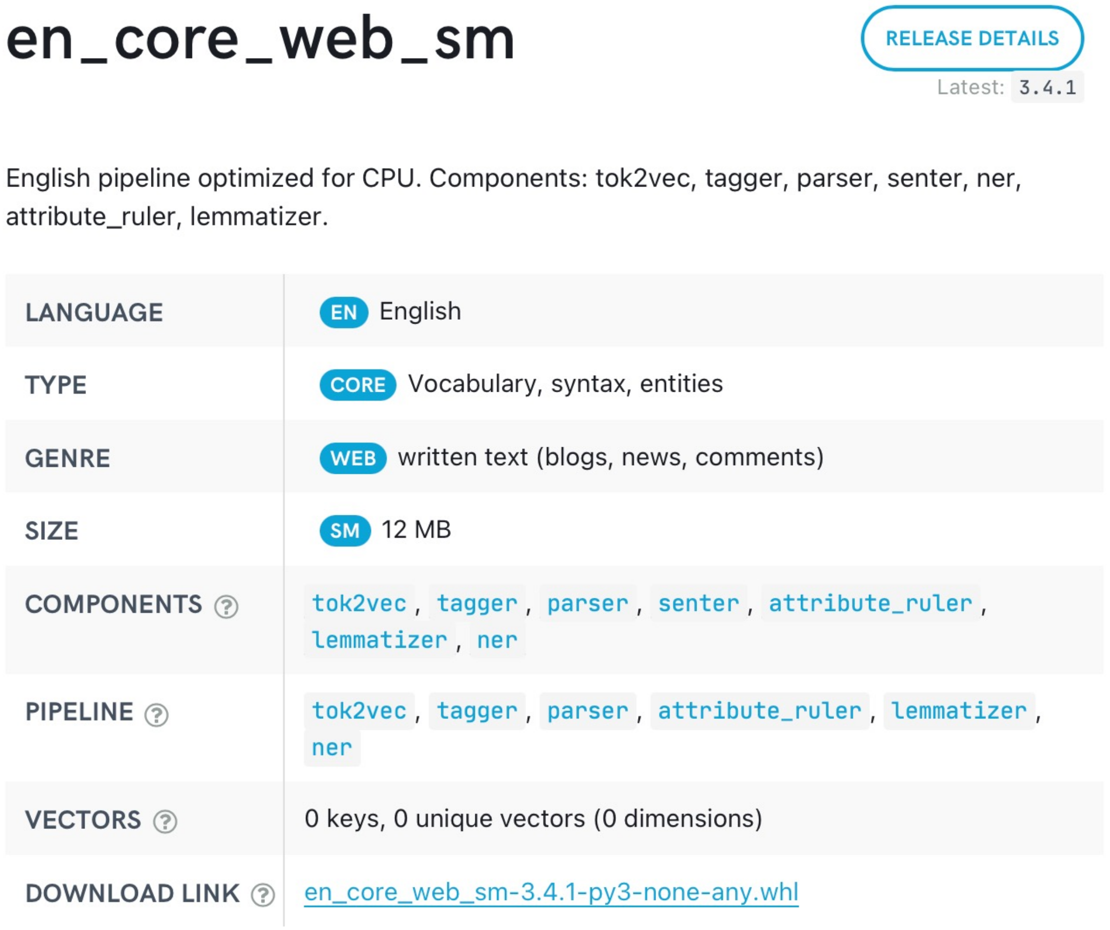
- Do you think the order of any of these components in the pipeline matter?



NAME	COMPONENT	CREATES	DESCRIPTION
tokenizer	`Tokenizer`	`Doc`	Segment text into tokens.
processing pipeline			
tagger	`Tagger`	`Token.tag`	Assign part-of-speech tags.
parser	`DependencyParser`	`Token.head`, `Token.dep`, `Doc.sents`, `Doc.noun_chunks`	Assign dependency labels.
ner	`EntityRecognizer`	`Doc.ents`, `Token.ent_iob`, `Token.ent_type`	Detect and label named entities.
lemmatizer	`Lemmatizer`	`Token.lemma`	Assign base forms.
textcat	`TextCategorizer`	`Doc.cats`	Assign document labels.
custom	custom components	`Doc._.xxx`, `Token._.xxx`, `Span._.xxx`	Assign custom attributes, methods or properties.

SpaCy has good documentation

- https://spacy.io/models/en



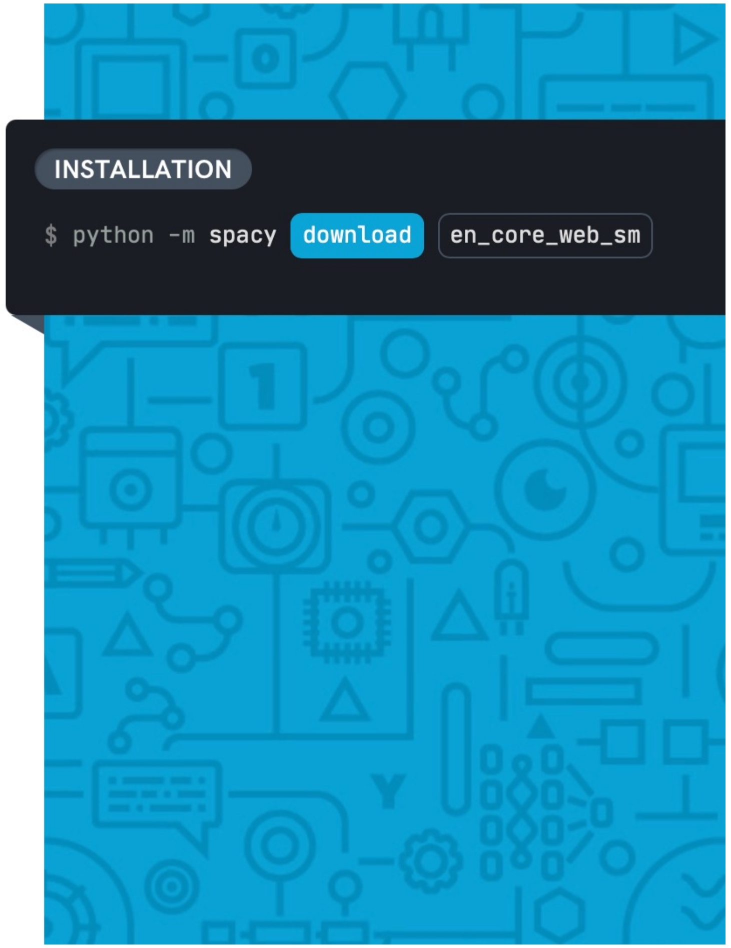
en_core_web_sm

RELEASE DETAILS

Latest: 3.4.1

English pipeline optimized for CPU. Components: tok2vec, tagger, parser, senter, ner, attribute_ruler, lemmatizer.

LANGUAGE	EN English
TYPE	CORE Vocabulary, syntax, entities
GENRE	WEB written text (blogs, news, comments)
SIZE	SM 12 MB
COMPONENTS ?	`tok2vec`, `tagger`, `parser`, `senter`, `attribute_ruler`, `lemmatizer`, `ner`
PIPELINE ?	`tok2vec`, `tagger`, `parser`, `attribute_ruler`, `lemmatizer`, `ner`
VECTORS ?	0 keys, 0 unique vectors (0 dimensions)
DOWNLOAD LINK ?	en_core_web_sm-3.4.1-py3-none-any.whl



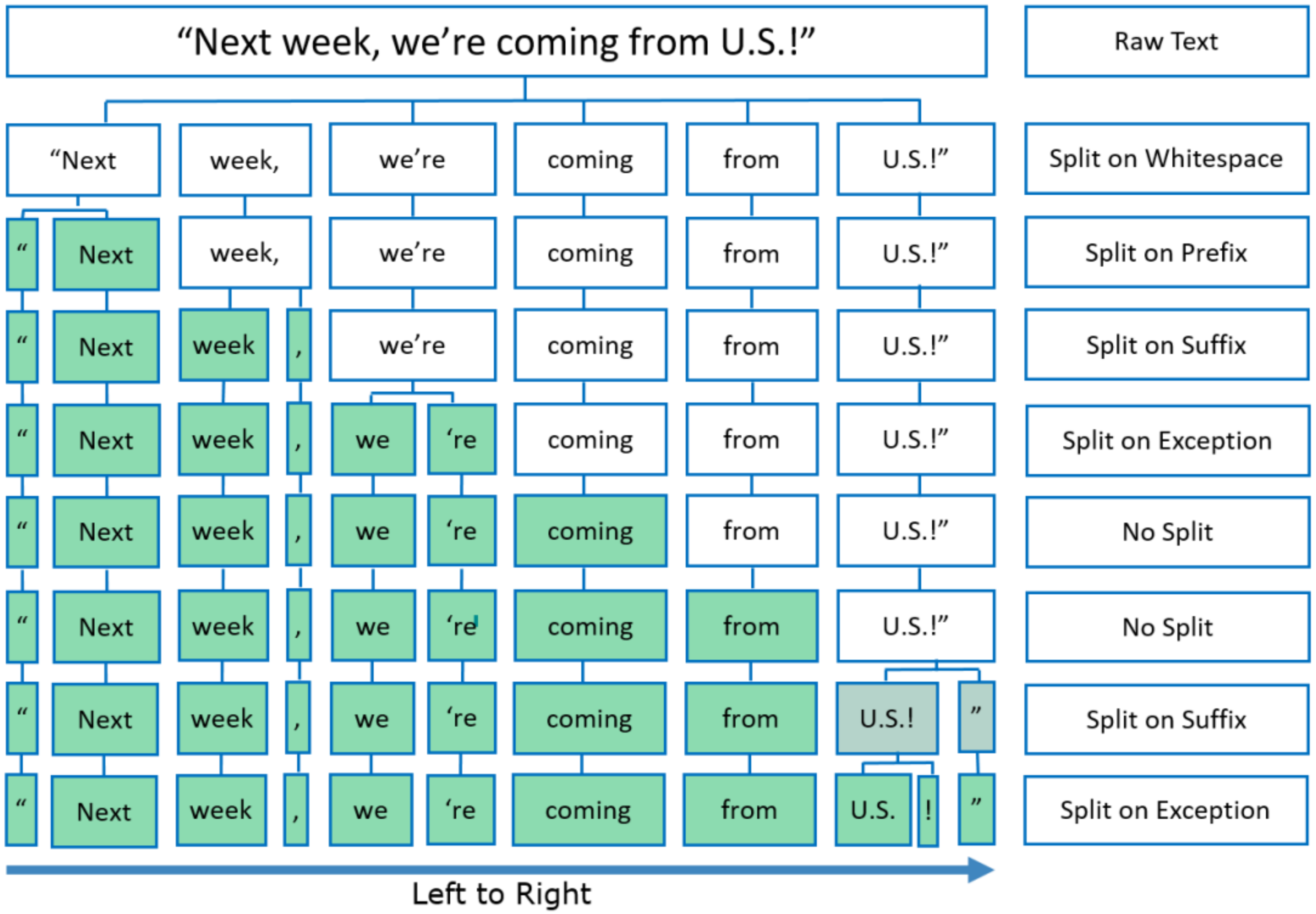
INSTALLATION

```
$ python -m spacy download en_core_web_sm
```

SpaCy contains only the best-suited algorithm for a problem in its toolkit

1. Collect text data in the target language
2. Annotate the text data with linguistic information such as part-of-speech tags, dependencies, and named entities
3. Preprocess the annotated data
4. In SpaCy, you can use the *train* command to train a new language model using your annotated data

Tokenization using spaCy



SpaCy lemmatization

- SpaCy supports lemmatization and does not have tools for stemming
- Also based on Wordnet
- Both NLTK and SpaCy use a look-up approach

What if I am not working with English?

```
# Install data for the French language for NLTK
nltk.download('fr’)
# Install the WordNet lemmatizer for the French language
nltk.download('wordnet_lemmatizer_fr')
```

```
# install French language model for SpaCy
!python -m spacy download fr_core_news_sm
# Load the French model
nlp = spacy.load("fr_core_news_sm")
```

Can you take tools from different models and put them in the same pipeline?

- To some extent
- Let’s build our own custom pipeline for different taggers

Activity

- Find out how many stop words NLTK has
- Find out how many stop words SpaCy has
- How similar are the lists? What makes them different?

Activity

- Any differences between SpaCy and NLTK Lemmatization?
- Were you able to show the difference between stemming and lemmatization?

Next time:

- Final NLTK and SpaCy comparisons
- Get started with BeautifulSoup?