Computational Linguistics Seminar
spaCy NLP

Marc Verhagen
Brandeis University
Spring 2021

The banner image is a fragment of Primordial Soup at https://regenaxe.com/2017/01/17/primordial-soup/

Overview

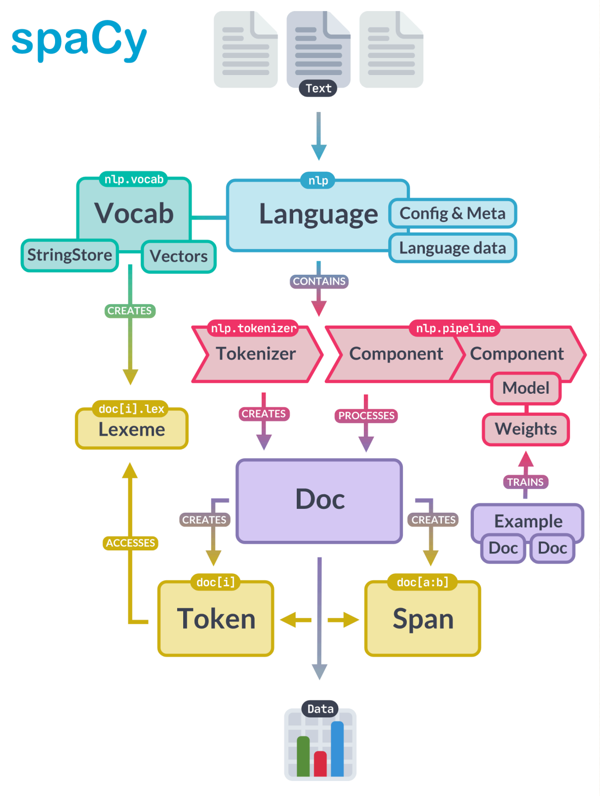
- Assignment
- Some spaCy concepts
  - https://spacy.io/usage/spacy-101
- Pattern Matching
  - token patterns and phrase patterns
- Named Entities
- Vectors and similarities

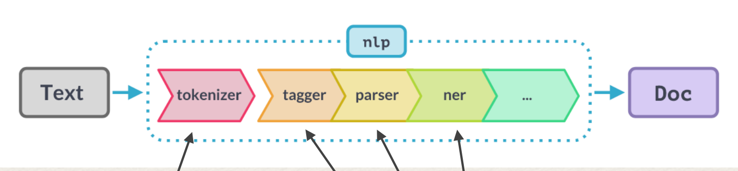
spaCy concepts

- Architecture
  - Text, documents, pipelines and annotations
  - The vocabulary: Tokens versus Lexemes
- Doc, Span and Token

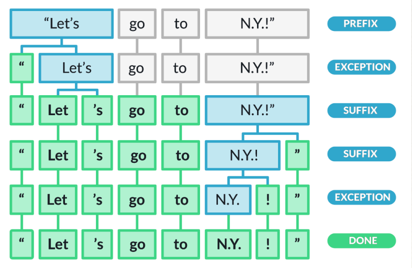
Architecture

Images are from
`https://spacy.io/usage/spacy-101`

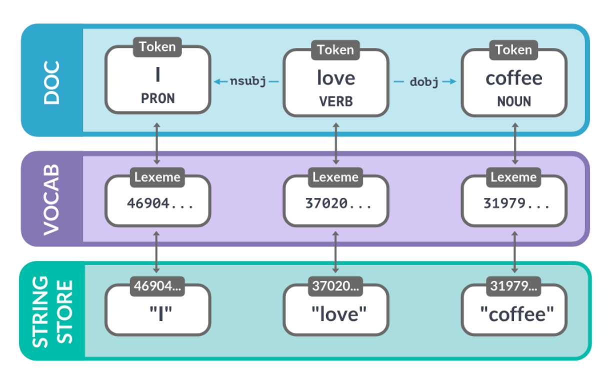




convolutional neural network models



rule-based system



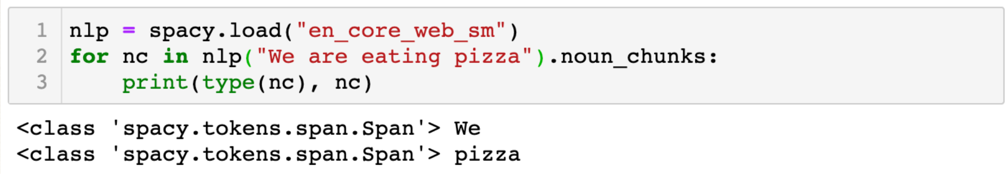
Docs, Spans and Tokens

- Documents hold all annotations
- Tokens are created by the tokenizer when the document is first created
- Other components add to the Doc or Token object

name	**description**	**creates**
tagger	Part-of-speech tagger	Token.tag, Token.pos
parser	Dependency parser	Token.dep, Token.head, Doc.sents, Doc.noun_chunks
ner	Named entity recognizer	Doc.ents, Token.ent_iob, Token.ent_type

Docs, Spans and Tokens

- Spans are slices of documents
- When you loop over them…
  - … you get tokens (just like with the document)
- Named entities, sentences and noun chunks:
  - are all stored on the Document
  - are all instances of Span

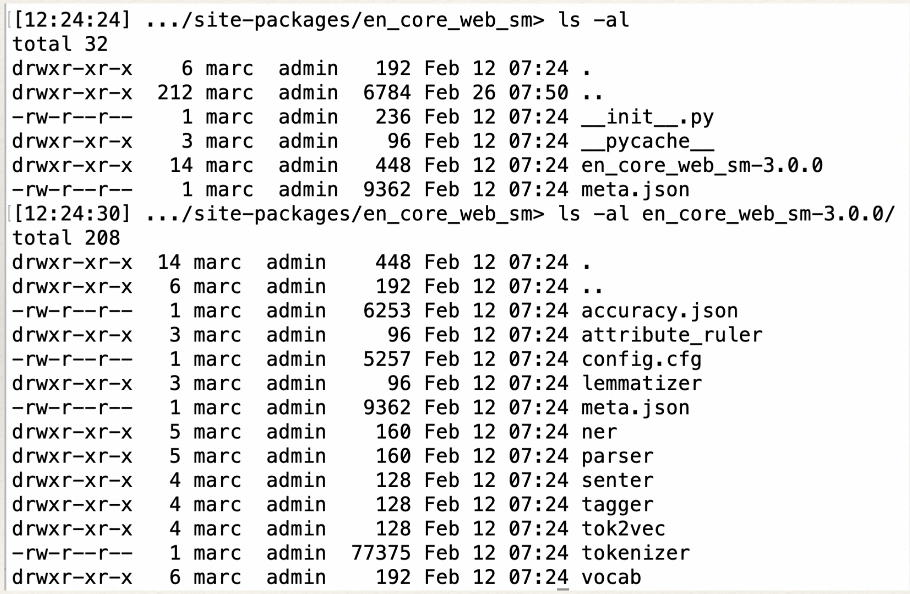


```
nlp = spacy.load("en_core_web_sm")
for nc in nlp("We are eating pizza").noun_chunks:
    print(type(nc), nc)
```

```
<class 'spacy.tokens.span.Span'> We
<class 'spacy.tokens.span.Span'> pizza
```

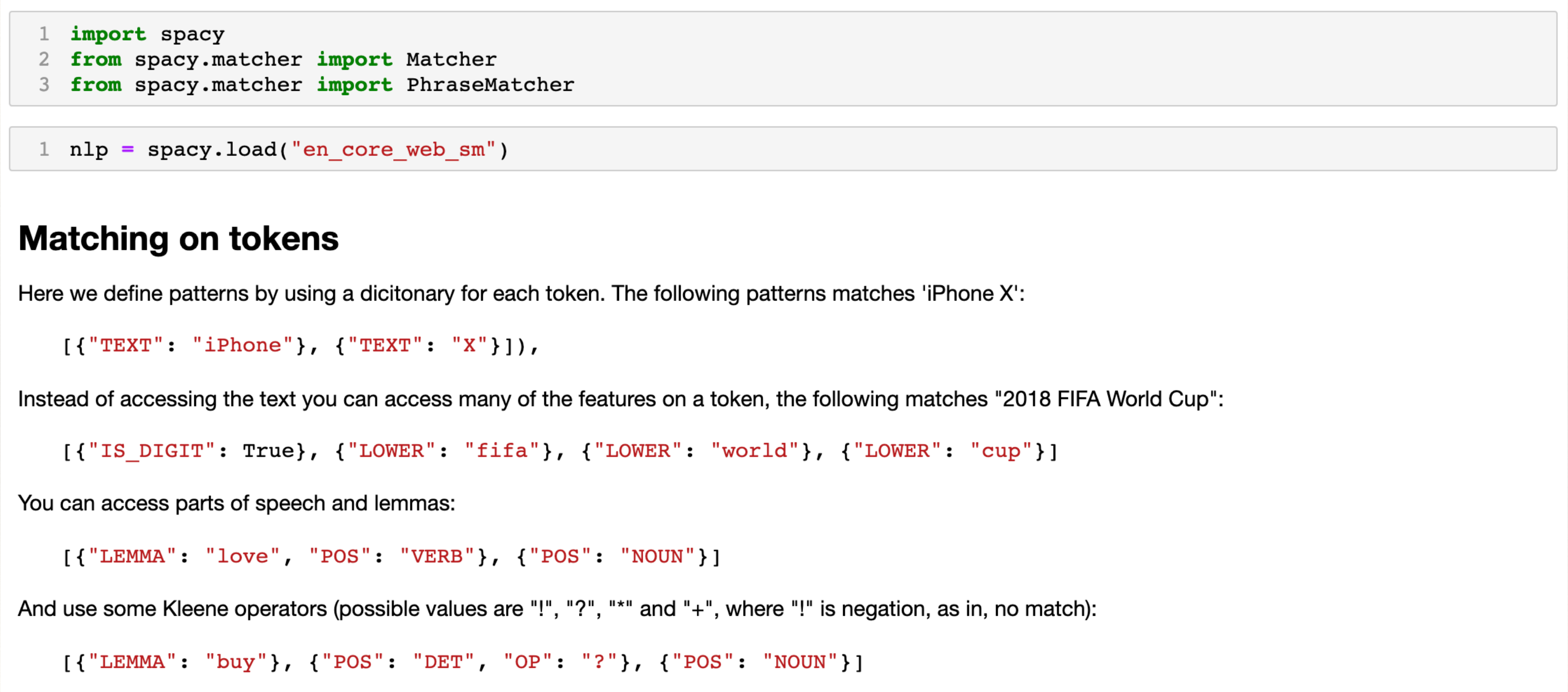
Downloading a model

- $> python -m spacy download en_core_web_sm



```
[12:24:24] .../site-packages/en_core_web_sm> ls -al
total 32
drwxr-xr-x    6 marc  admin   192 Feb 12 07:24 .
drwxr-xr-x  212 marc  admin  6784 Feb 26 07:50 ..
-rw-r--r--    1 marc  admin   236 Feb 12 07:24 __init__.py
drwxr-xr-x    3 marc  admin    96 Feb 12 07:24 __pycache__
drwxr-xr-x   14 marc  admin   448 Feb 12 07:24 en_core_web_sm-3.0.0
-rw-r--r--    1 marc  admin  9362 Feb 12 07:24 meta.json
[12:24:30] .../site-packages/en_core_web_sm> ls -al en_core_web_sm-3.0.0/
total 208
drwxr-xr-x  14 marc  admin    448 Feb 12 07:24 .
drwxr-xr-x   6 marc  admin    192 Feb 12 07:24 ..
-rw-r--r--   1 marc  admin   6253 Feb 12 07:24 accuracy.json
drwxr-xr-x   3 marc  admin     96 Feb 12 07:24 attribute_ruler
-rw-r--r--   1 marc  admin   5257 Feb 12 07:24 config.cfg
drwxr-xr-x   3 marc  admin     96 Feb 12 07:24 lemmatizer
-rw-r--r--   1 marc  admin   9362 Feb 12 07:24 meta.json
drwxr-xr-x   5 marc  admin    160 Feb 12 07:24 ner
drwxr-xr-x   5 marc  admin    160 Feb 12 07:24 parser
drwxr-xr-x   4 marc  admin    128 Feb 12 07:24 senter
drwxr-xr-x   4 marc  admin    128 Feb 12 07:24 tagger
drwxr-xr-x   4 marc  admin    128 Feb 12 07:24 tok2vec
-rw-r--r--   1 marc  admin  77375 Feb 12 07:24 tokenizer
drwxr-xr-x   6 marc  admin    192 Feb 12 07:24 vocab
```

Pattern Matching



```python
import spacy
from spacy.matcher import Matcher
from spacy.matcher import PhraseMatcher
```

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

Matching on tokens

Here we define patterns by using a dicitonary for each token. The following patterns matches 'iPhone X':

```
[{"TEXT": "iPhone"}, {"TEXT": "X"}]),
```

Instead of accessing the text you can access many of the features on a token, the following matches "2018 FIFA World Cup":

```
[{"IS_DIGIT": True}, {"LOWER": "fifa"}, {"LOWER": "world"}, {"LOWER": "cup"}]
```

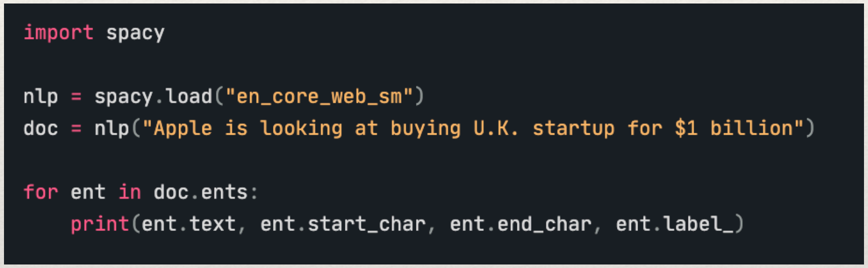
You can access parts of speech and lemmas:

```
[{"LEMMA": "love", "POS": "VERB"}, {"POS": "NOUN"}]
```

And use some Kleene operators (possible values are "!", "?", "*" and "+", where "!" is negation, as in, no match):

```
[{"LEMMA": "buy"}, {"POS": "DET", "OP": "?"}, {"POS": "NOUN"}]
```

Named Entities

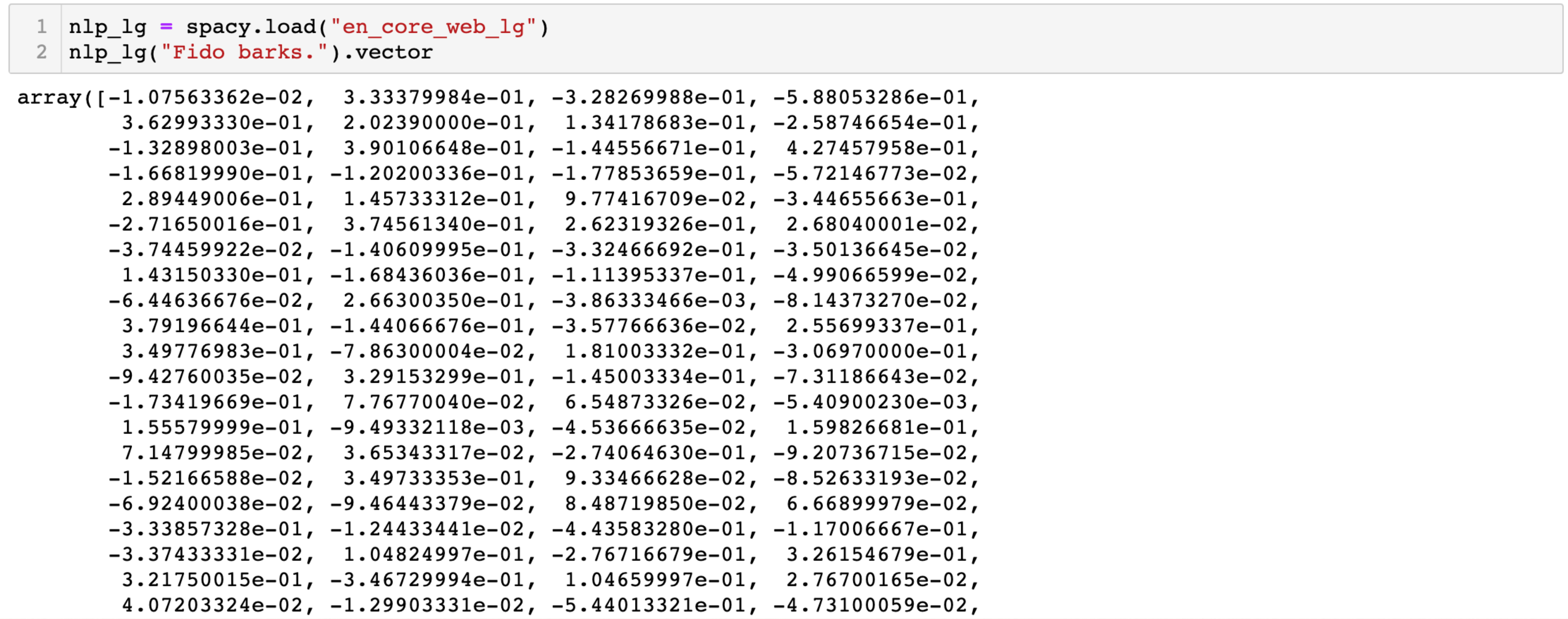


```python
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("Apple is looking at buying U.K. startup for $1 billion")

for ent in doc.ents:
    print(ent.text, ent.start_char, ent.end_char, ent.label_)
```

Similarities



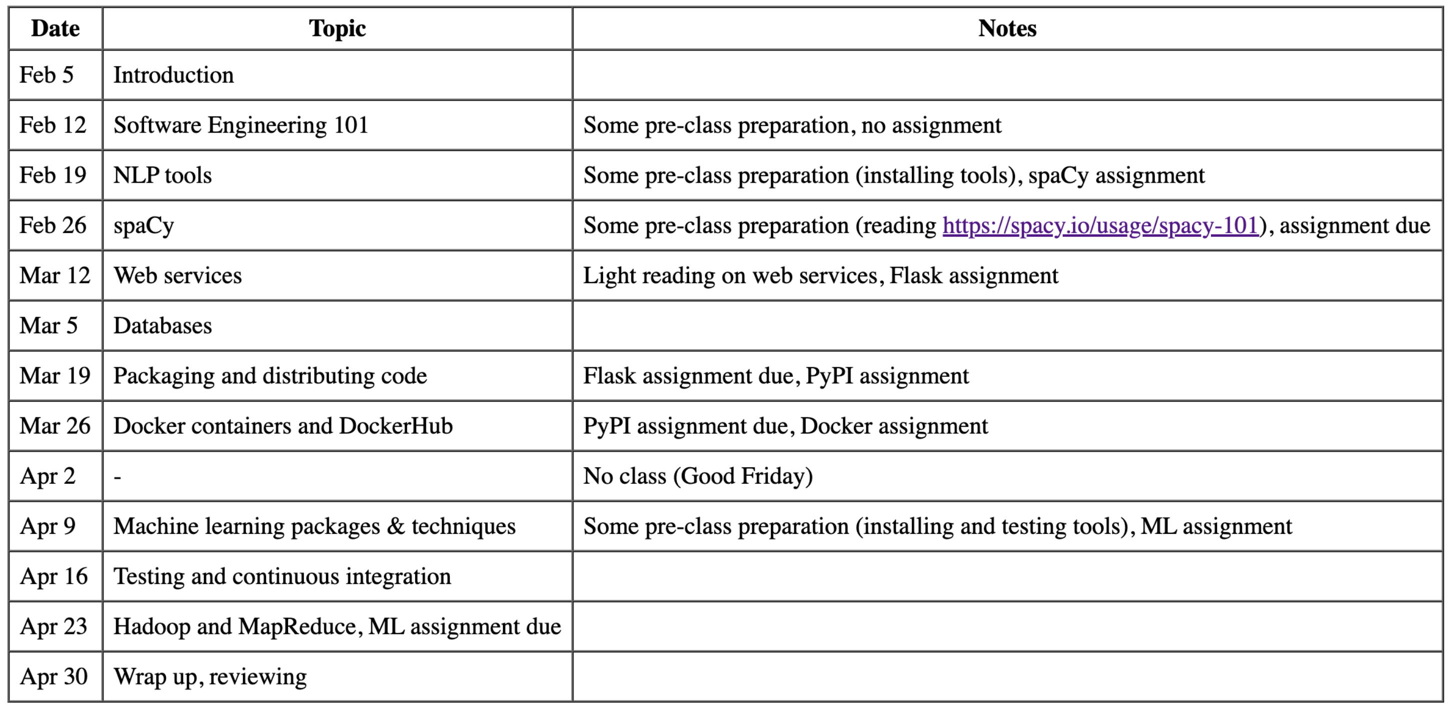
```python
nlp_lg = spacy.load("en_core_web_lg")
nlp_lg("Fido barks.").vector
```

```
array([-1.07563362e-02,  3.33379984e-01, -3.28269988e-01, -5.88053286e-01,
        3.62993330e-01,  2.02390000e-01,  1.34178683e-01, -2.58746654e-01,
       -1.32898003e-01,  3.90106648e-01, -1.44556671e-01,  4.27457958e-01,
       -1.66819990e-01, -1.20200336e-01, -1.77853659e-01, -5.72146773e-02,
        2.89449006e-01,  1.45733312e-01,  9.77416709e-02, -3.44655663e-01,
       -2.71650016e-01,  3.74561340e-01,  2.62319326e-01,  2.68040001e-02,
       -3.74459922e-02, -1.40609995e-01, -3.32466692e-01, -3.50136645e-02,
        1.43150330e-01, -1.68436036e-01, -1.11395337e-01, -4.99066599e-02,
       -6.44636676e-02,  2.66300350e-01, -3.86333466e-03, -8.14373270e-02,
        3.79196644e-01, -1.44066676e-01, -3.57766636e-02,  2.55699337e-01,
        3.49776983e-01, -7.86300004e-02,  1.81003332e-01, -3.06970000e-01,
       -9.42760035e-02,  3.29153299e-01, -1.45003334e-01, -7.31186643e-02,
       -1.73419669e-01,  7.76770040e-02,  6.54873326e-02, -5.40900230e-03,
        1.55579999e-01, -9.49332118e-03, -4.53666635e-02,  1.59826681e-01,
        7.14799985e-02,  3.65343317e-02, -2.74064630e-01, -9.20736715e-02,
       -1.52166588e-02,  3.49733353e-01,  9.33466628e-02, -8.52633193e-02,
       -6.92400038e-02, -9.46443379e-02,  8.48719850e-02,  6.66899979e-02,
       -3.33857328e-01, -1.24433441e-02, -4.43583280e-01, -1.17006667e-01,
       -3.37433331e-02,  1.04824997e-01, -2.76716679e-01,  3.26154679e-01,
        3.21750015e-01, -3.46729994e-01,  1.04659997e-01,  2.76700165e-02,
        4.07203324e-02, -1.29903331e-02, -5.44013321e-01, -4.73100059e-02,
```

Mini Presentation

- Think of which topic really interests you
  - Does not have to be on the schedule, but should be related to the seminar
  - Possible topics
    - dive into textblob or polyglot
    - what is available for continuous integration
    - noSQL databases
- Prepare to talk about it for 5-10 minutes
- Contact me

Schedule



Date	Topic	Notes
Feb 5	Introduction	
Feb 12	Software Engineering 101	Some pre-class preparation, no assignment
Feb 19	NLP tools	Some pre-class preparation (installing tools), spaCy assignment
Feb 26	spaCy	Some pre-class preparation (reading https://spacy.io/usage/spacy-101), assignment due
Mar 12	Web services	Light reading on web services, Flask assignment
Mar 5	Databases	
Mar 19	Packaging and distributing code	Flask assignment due, PyPI assignment
Mar 26	Docker containers and DockerHub	PyPI assignment due, Docker assignment
Apr 2	-	No class (Good Friday)
Apr 9	Machine learning packages & techniques	Some pre-class preparation (installing and testing tools), ML assignment
Apr 16	Testing and continuous integration	
Apr 23	Hadoop and MapReduce, ML assignment due	
Apr 30	Wrap up, reviewing	

https://marcverhagen.github.io/CS138A/