

Computational Linguistics Seminar

Generic NLP Tools

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

❖ NLTK & TextBlob



❖ Polyglot



❖ spaCy



❖ Assignments

❖ Not yet: gensim, sklearn, PyTorch, TensorFlow, BERT

TextBlob

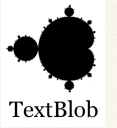


- ❖ <https://textblob.readthedocs.io/en/dev/>
- ❖ Build on top of NLTK and pattern
 - ❖ <http://www.nltk.org/>
 - ❖ The granddaddy of all Python NLP tools
- ❖ But also on Pattern
 - ❖ <https://pypi.org/project/Pattern/>
 - ❖ <https://stackabuse.com/python-for-nlp-introduction-to-the-pattern-library/>
 - ❖ Library for NLP tasks (tokenization, stemming, POS tagging, sentiment analysis), Data Mining (APIs to Twitter, Facebook, Wikipedia) and Machine Learning (SVM, KNN, perceptron).
 - ❖ Home page is down, no active support.



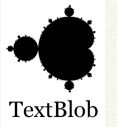
Pattern 3.6

TextBlob



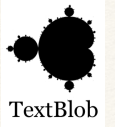
- ❖ Not industrial strength
- ❖ Easy to learn and good for prototyping

TextBlob



- ❖ Tokenization and Part-of-speech tagging
- ❖ Word inflection (pluralization and singularization) and lemmatization
- ❖ Spelling correction
- ❖ Noun phrase extraction and shallow parsing
- ❖ WordNet integration
- ❖ Word and phrase frequencies, n-grams

TextBlob



- ❖ **Sentiment analysis**

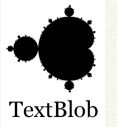
- ❖ Sentiment

- ❖ Depending upon the most commonly occurring positive (good, best, excellent, etc.) and negative (bad, awful, pathetic, etc.) adjectives, a sentiment score between 1 and -1 is assigned to the text. This sentiment score is also called the polarity.

- ❖ Subjectivity

- ❖ Quantifies the amount of personal opinion and factual information contained in the text. The higher subjectivity means that the text contains personal opinion rather than factual information. A value between 0 and 1.

TextBlob



- ❖ **Classification** (Naive Bayes, Decision Tree, MaxEnt)
 - ❖ <https://textblob.readthedocs.io/en/dev/classifiers.html>
 - ❖ Wrapper around NLTK classes
 - ❖ Slightly simpler to use
 - ❖ Quick and dirty stuff only, use sklearn or another library for serious classification.

TextBlob



❖ Extensions

- ❖ Add new models or languages through extensions
- ❖ Limited availability of extensions:

textblob-fr	Frech
textblob-de	German
textblob-aptagger	A fast and accurate tagger based on the Averaged Perceptron

Polyglot



- ❖ <https://polyglot.readthedocs.io/en/latest/>
- ❖ <https://github.com/aboSamoor/polyglot>

Feature	Languages
tokenization	165
language detection	196
named entity recognition	40
POS tagging	16
sentiment analysis	136
word embeddings	137
morphological analysis	135
transliteration	69

❖ Language detection

```
>>> from polyglot.detect import Detector
>>> arabic_text = u"""
... أفاد مصدر امني في قيادة عمليات صلاح الدين في العراق بأن " القوات الامنية تتوقف لليوم
... الثالث على التوالي عن التقدم الى داخل مدينة تكريت بسبب
... انتشار قناصي التنظيم الذي يطلق على نفسه اسم "الدولة الاسلامية" والعبوات الناسفة
... ". والمنازل المفخخة والانتحاريين، فضلا عن ان القوات الامنية تنتظر وصول تعزيزات اضافية
... """
>>> detector = Detector(arabic_text)
>>> print(detector.language)
name: Arabic          code: ar          confidence: 99.0 read bytes: 907
```


❖ Language detection (mixed)

```
>>> mixed_text = u"""
... China (simplified Chinese: 中国; traditional Chinese: 中國),
... officially the People's Republic of China (PRC), is a sovereign state
... located in East Asia.
... """
>>> for language in Detector(mixed_text).languages:
...     print(language)
...
name: English      code: en      confidence: 87.0 read bytes: 1154
name: Chinese      code: zh_Hant confidence:  5.0 read bytes: 1755
name: un           code: un           confidence:  0.0 read bytes:  0
```

❖ Tokenization and segmentation

```
>>> from polyglot.text import Text
>>> blob = u""导致4人死亡，据悉这起事件与法国《查理周刊》杂志社恐怖袭击案有关。""
>>> text = Text(blob)
WordList(['导致', '4', '人', '死亡', ',', ' ', '据悉', '这', '起', '事件', '与', '法', '国', ' ', '《', '查理', '周刊', '》', '杂志', '社', '恐怖', '袭击', '案', '有关', '。'])
>>> text.sentences
[Sentence("导致4人死亡，据悉这起事件与法国《查理周刊》杂志社恐怖袭击案有关。")]
```


Polyglot



❖ Morphology

❖ Need to download models first

❖ <https://polyglot.readthedocs.io/en/latest/MorphologicalAnalysis.html>

```
>>> from polyglot.text import Text
>>> words = ["preprocessing", "processor", "invaluable", "thankful", "crossed"]
>>> for w in words:
...     w = Word(w, language="en")
...     print("{:<20}{}".format(w, w.morphemes))
preprocessing      ['pre', 'process', 'ing']
processor          ['process', 'or']
invaluable         ['in', 'valuable']
thankful           ['thank', 'ful']
crossed            ['cross', 'ed']
```

Polyglot



❖ Named entities

```
(base) root@67f9b50bcc31:/app# polyglot download embeddings2.en ner2.en
[polyglot_data] Downloading package embeddings2.en to
[polyglot_data]      /root/polyglot_data...
[polyglot_data] Downloading package ner2.en to /root/polyglot_data...
```

```
>>> from polyglot.text import Text
>>> blob = """The Israeli Prime Minister Benjamin Netanyahu has warned that Iran
poses a "threat to the entire world"."""
>>> text = Text(blob)
>>> for sent in text.sentences:
...     for entity in sent.entities:
...         print(entity.tag, entity)
I-ORG ['Israeli']
I-PER ['Benjamin', 'Netanyahu']
I-LOC ['Iran']
```


Polyglot



❖ Sentiment (just polarity per word really)

```
(base) root@67f9b50bcc31:/app# polyglot download sentiment2.en
[polyglot_data] Downloading package sentiment2.en to
[polyglot_data]      /root/polyglot_data...
```

```
>>> from polyglot.text import Text
>>> from polyglot.text import Text
>>> text = Text("The movie was really good.")
>>> for w in text.words:
...     print("{:<16}{:>2}".format(w, w.polarity))
...
The                0
movie              0
was                0
really             0
good               1
.                  0
```

spaCy



- ❖ The new gold standard in Python NLP processing
- ❖ Fast, lots of functionality, well-documented, well-tested
- ❖ Trainable and extendable
- ❖ Support for custom models in PyTorch, TensorFlow and other frameworks
- ❖ Comes with its own visualization module

TextBlob, Polyglot and spaCy compared

Feature	TextBlob	Polyglot	spaCy
Tokenization	✓	✓	
POS tagging	✓	✓	
Morphology - lemmas	✓		
Morphology - morphemes		✓	
Morphology - inflection	✓		
NP extraction	✓		
Shallow parsing	✓		
Full parsing			
Dependency parsing			

TextBlob, Polyglot and spaCy compared

Feature	TextBlob	Polyglot	spaCy
Named entity extraction	✓	✓	
Sentiment analysis	✓	✓	
Spell checking	✓		
Word and phrase frequencies, n-grams	✓		
Word embeddings		✓	
Classification	✓		
Multiple languages	✓	✓✓✓	
Language detection		✓	
Transliteration		✓	

Assignments

- ❖ Create a simple spaCy tool to markup some text (this week)
- ❖ Future assignments
 - ❖ Make the tool accessible via a RESTful service (Flask)
 - ❖ Create a web interface (Flask)
 - ❖ Make the tool available via PyPI
 - ❖ Dockerize the tool
 - ❖ Machine Learning assignment

Tool accepts plain text

When Sebastian Thrun started working on self-driving cars at Google in 2007, few people outside of the company took him seriously. “I can tell you very senior CEOs of major American car companies would shake my hand and turn away because I wasn’t worth talking to,” said Thrun, in an interview with Recode earlier this week.

Tool generates markup

```
<markup>When <entity class="PERSON">Sebastian Thrun</entity> started  
working on self-driving cars at Google in <entity class="DATE">2007</entity>,  
few people outside of the company took him seriously. "I can tell you very  
senior CEOs of major <entity class="NORP">American</entity> car companies  
would shake my hand and turn away because I wasn't worth talking to," said  
<entity class="PERSON">Thrun</entity>, in an interview with <entity  
class="PERSON">Recode</entity> <entity class="DATE">earlier this week</  
entity>.</markup>
```

Tool generates markup

```
>>> import ner
>>> text = open('input.txt').read()
>>> text
'When Sebastian Thrun started working on self-driving cars at Google in 2007, few
people outside of the company took him seriously. "I can tell you very senior CEOs
of major American car companies would shake my hand and turn away because I wasn't
worth talking to," said Thrun, in an interview with Recode earlier this week.\n'
>>> ner.entity_markup(text)
'<markup>When <entity class="PERSON">Sebastian Thrun</entity> started working on
self-driving cars at Google in <entity class="DATE">2007</entity>, few people
outside of the company took him seriously. "I can tell you very senior CEOs of
major <entity class="NORP">American</entity> car companies would shake my hand and
turn away because I wasn't worth talking to," said <entity class="PERSON">Thrun</
entity>, in an interview with <entity class="PERSON">Recode</entity> <entity
class="DATE">earlier this week</entity>.\n</markup>'
>>>
```

RESTful service

GET request returns some descriptive JSON

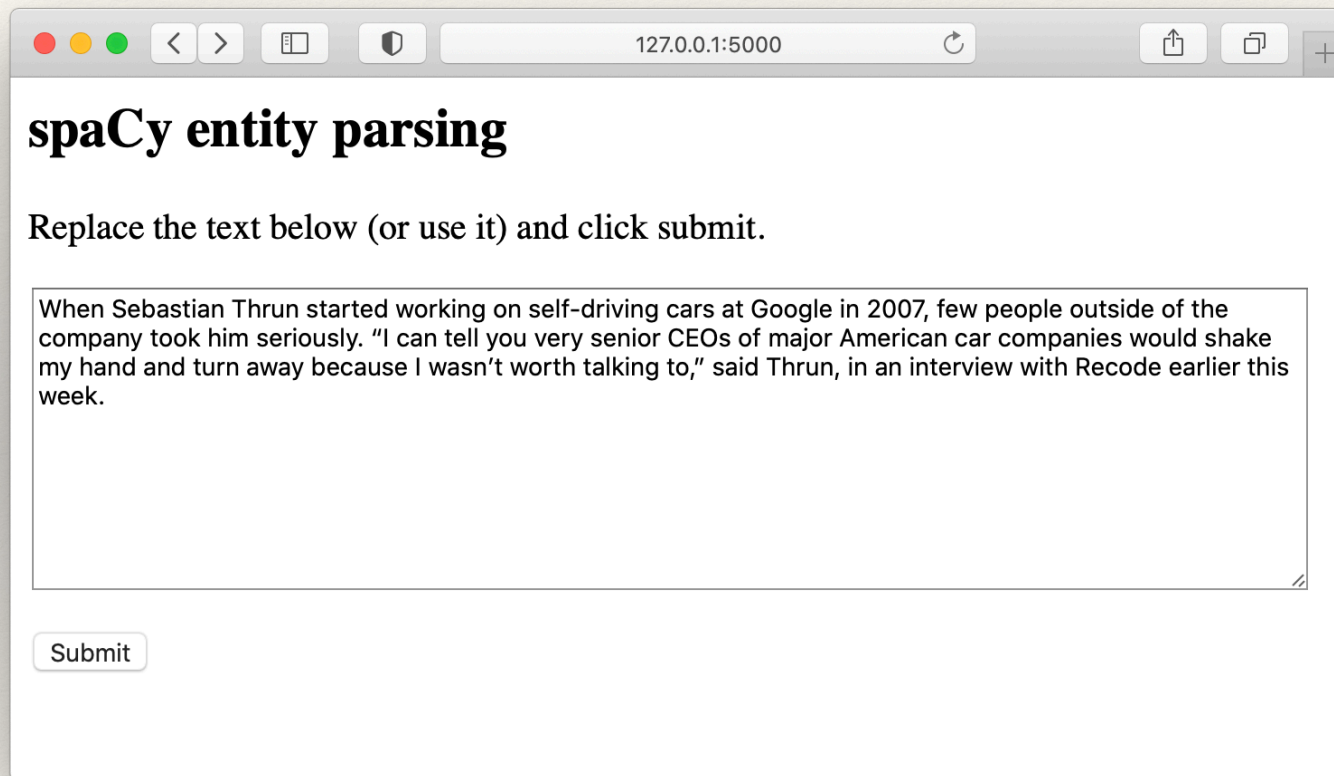
```
$> curl http://127.0.0.1:5000/api
{
  "description": "Interface to the spaCy entity extractor",
  "usage": "curl -X POST -d@input.txt http://127.0.0.1:5000/"
}
```

RESTful service

POST request returns JSON with markup

```
$> curl -X POST -d@input.txt http://127.0.0.1:5000/api
{
  "input": "When Sebastian Thrun started working on self-driving cars at Google
in 2007, few people outside of the company took him seriously. \u201cI can tell
you very senior CEOs of major American car companies would shake my hand and turn
away because I wasn\u2019t worth talking to,\u201d said Thrun, in an interview
with Recode earlier this week.",
  "output": "<markup>When <entity class=\"PERSON\">Sebastian Thrun</entity>
started working on self-driving cars at Google in <entity class=\"DATE\">2007</
entity>, few people outside of the company took him seriously. \u201cI can tell
you very senior CEOs of major <entity class=\"NORP\">American</entity> car
companies would shake my hand and turn away because I wasn\u2019t worth talking
to,\u201d said <entity class=\"PERSON\">Thrun</entity>, in an interview with
<entity class=\"PERSON\">Recode</entity> <entity class=\"DATE\">earlier this
week</entity>.</markup>"
}
```


Web Interface



The image shows a web browser window with a title bar containing standard macOS window controls (red, yellow, green buttons) and navigation icons (back, forward, home, refresh). The address bar displays the URL "127.0.0.1:5000". The main content area has a heading "spaCy entity parsing" in a bold, black, serif font. Below the heading is a text prompt: "Replace the text below (or use it) and click submit." Underneath this prompt is a large, empty text input field with a thin gray border. At the bottom left of the page, there is a "Submit" button with a rounded rectangular shape and a light gray background.

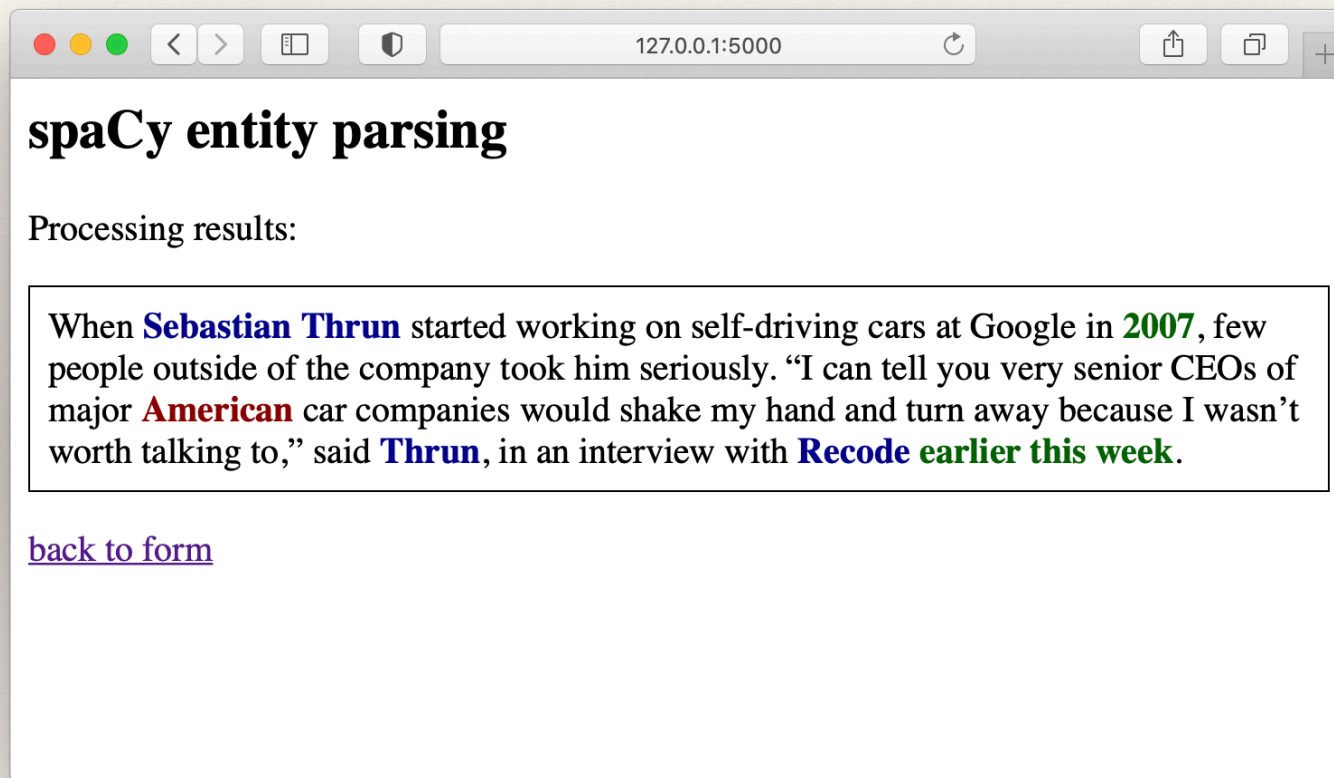
spaCy entity parsing

Replace the text below (or use it) and click submit.

When Sebastian Thrun started working on self-driving cars at Google in 2007, few people outside of the company took him seriously. "I can tell you very senior CEOs of major American car companies would shake my hand and turn away because I wasn't worth talking to," said Thrun, in an interview with Recode earlier this week.

Submit

Web Interface



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	Machine learning packages & techniques	Some pre-class preparation (installing and testing tools), gensim or sklearn assignment
Mar 12	Machine learning packages & techniques	Some pre-class preparation (installing and testing tools), spaCy/PyTorch assignment
Mar 5	Databases	
Mar 19	Web services	
Mar 26	Packaging and distributing code	
Apr 2	-	No class (Good Friday)
Apr 9	Docker containers and DockerHub	
Apr 16	Testing and continuous integration	
Apr 23	Hadoop and MapReduce	
Apr 30	Wrap up, reviewing	