spaCy Documentation

Release 0.1

Matthew Honnibal

August 20, 2014

Contents

CHAPTER 1

Python API

1.1 Cheat Sheet

CHAPTER 2

Cython API

2.1 Cheat Sheet

CHAPTER 3

Adding a Language

CHAPTER 4

Overview

spaCy is a tokenizer for natural languages, tightly coupled to a global vocabulary store.

Instead of a list of strings, spaCy returns references to lexical types. All of the string-based features you might need are pre-computed for you:

```
>>> from spacy import en
>>> example = u"Apples aren't oranges..."
>>> apples, are, nt, oranges, ellipses = en.tokenize(example)
>>> en.is_punct(ellipses)
True
>>> en.get_string(en.word_shape(apples))
'Xxxx'
```

You also get lots of distributional features, calculated from a large sample of text:

```
>>> en.prob_of(are) > en.prob_of(oranges)
True
>>> en.can_noun(are)
False
>>> en.is_oft_title(apples)
False
```

CHAPTER 5

Pros and Cons

Pros:

- All tokens come with indices into the original string
- Full unicode support
- Extensible to other languages
- Batch operations computed efficiently in Cython
- Cython API
- numpy interoperability

Cons:

- It's new (released September 2014)
- Higher memory usage (up to 1gb)
- More conceptually complicated
- Tokenization rules expressed in code, not as data

CHAPTER 6

Installation

Installation via pip:

pip install spacy

From source, using virtualenv:

```
$ git clone http://github.com/honnibal/spaCy.git
$ cd spaCy
$ virtualenv .env
$ source .env/bin/activate
$ pip install -r requirements.txt
$ fab make
$ fab test
```