An introduction to spaCy

Industrial-strength Natural Language Processing in Python

Matthew Honnibal & Ines Montani 💥 **Explosion AI**

spaCy

- leading **free, open-source library** for Natural Language Processing (NLP) in Python
- designed specifically for **production use**
- helps you build applications that process and "understand" **large volumes of text**
- **use cases:** information extraction, language understanding, pre-process text for deep learning

What spaCy isn't

- a platform or "an API"
  spaCy is a **library** for building NLP systems, not a consumable service
- research software
  spaCy is built on the latest research, but designed **for production**. (Contrast: NLTK and CoreNLP, created for teaching and research.)

Overview

Important features

- Tokenization
- Part-of-speech (POS) tagging
- Dependency Parsing
- Sentence Boundary Detection (SBD)
- Named Entity Recognition (NER)
- Similarity
- Rule-based matching
- Training

Getting started


```
$  pip install spacy
$  python -m spacy download en
```

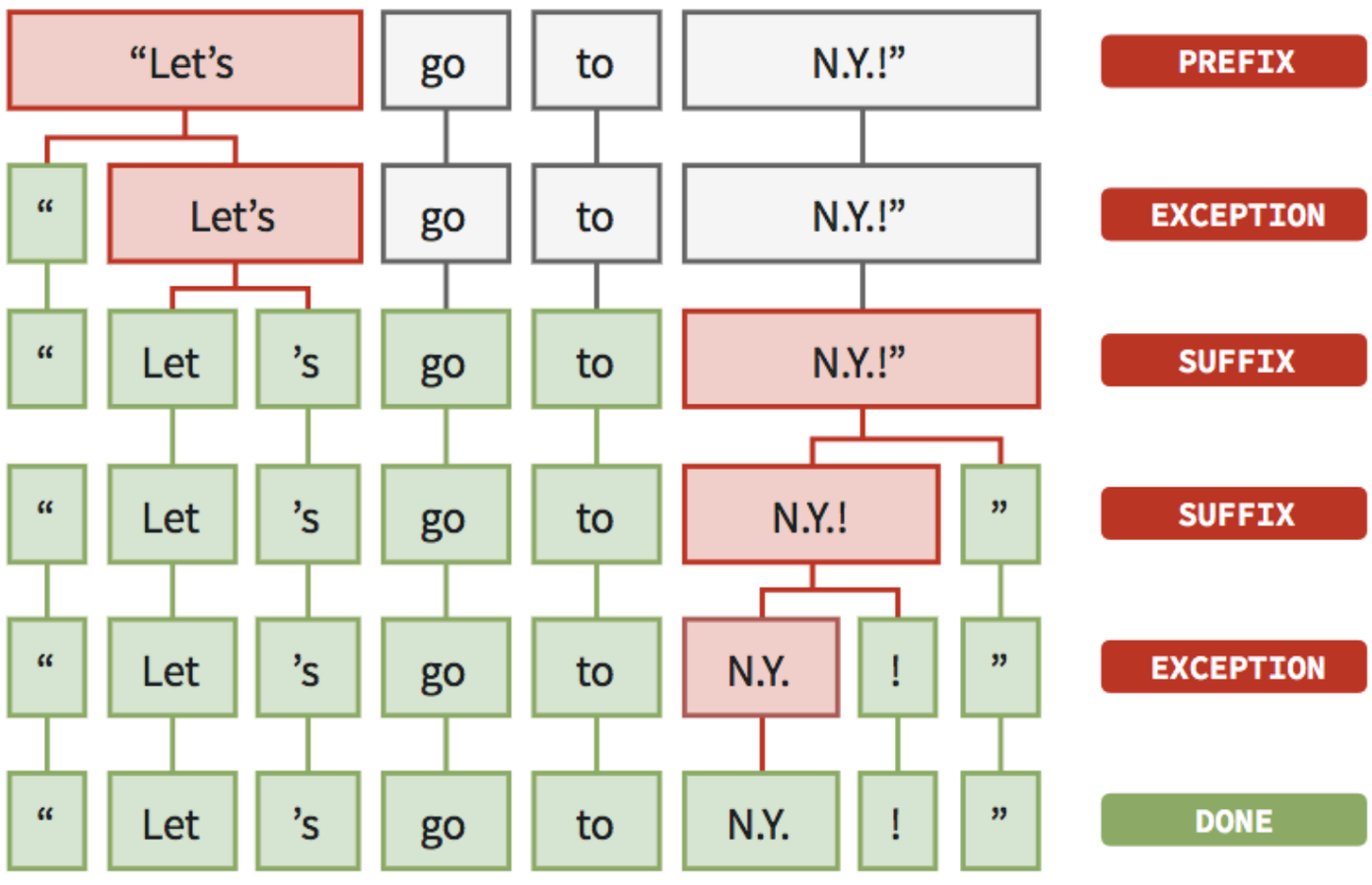
```
import spacy

nlp = spacy.load('en')
doc = nlp(u'Apple is looking at buying U.K. startup')
```

- models are required for features that need predictions, e.g. parsing, tagging or NER
- `spacy.load()` returns instance of `Language` containing all components and data needed to process text

Tokenization

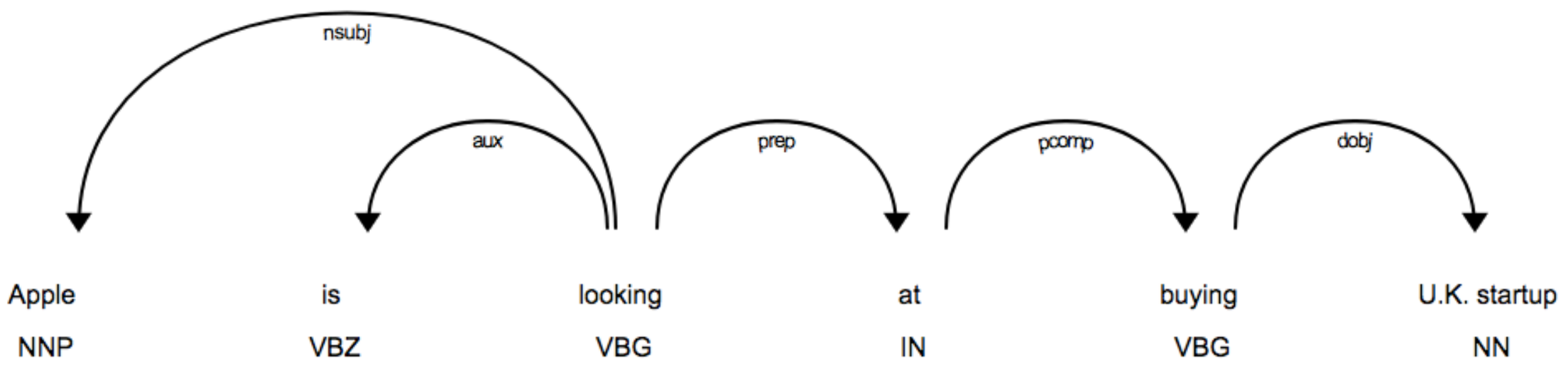
```
doc = nlp(u"Let's go to N.Y.!")
print([token.text for token in doc])
```



POS tags & dependencies


```
doc = nlp(u"Apple is looking at buying U.K. startup")

for token in doc:
    print(token.text, token.pos_, token.tag_)
```

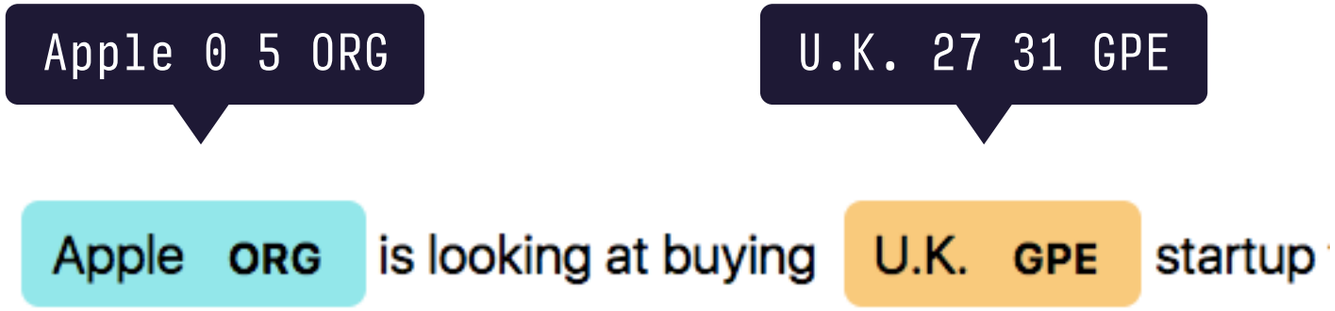


Named Entity Recognition

```
doc = nlp(u"Apple is looking at buying U.K. startup")

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

```
Apple 0 5 ORG
U.K. 27 31 GPE
```



- “real-world named objects”, e.g. person, country, product
- statistical, so requires **fine-tuning** depending on the data

Word vectors & similarity

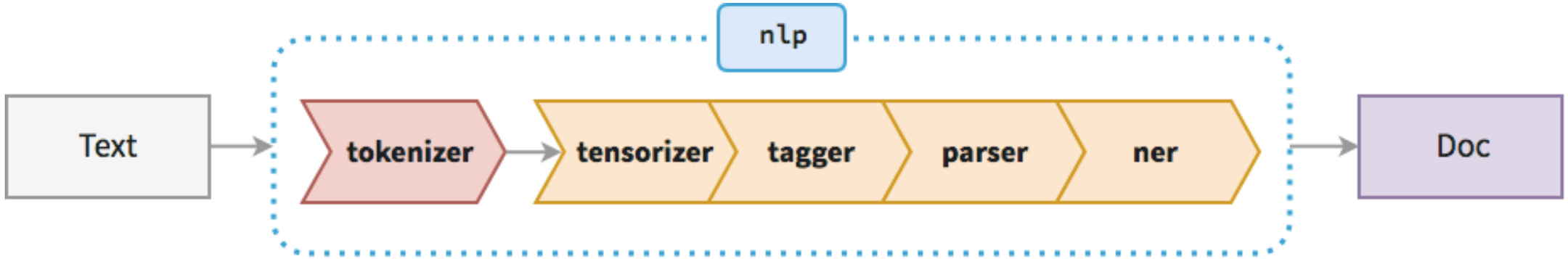

```
dog, cat, banana = nlp(u"dog cat banana")

dog.similarity(cat)  0.80
cat.similarity(dog)  0.80
dog.similarity(banana)  0.24
```

- compare two objects and make a prediction of **how similar they are**
- determined by comparing **word vectors**: multi-dimensional **meaning representations**
- generated using algorithm like Word2Vec

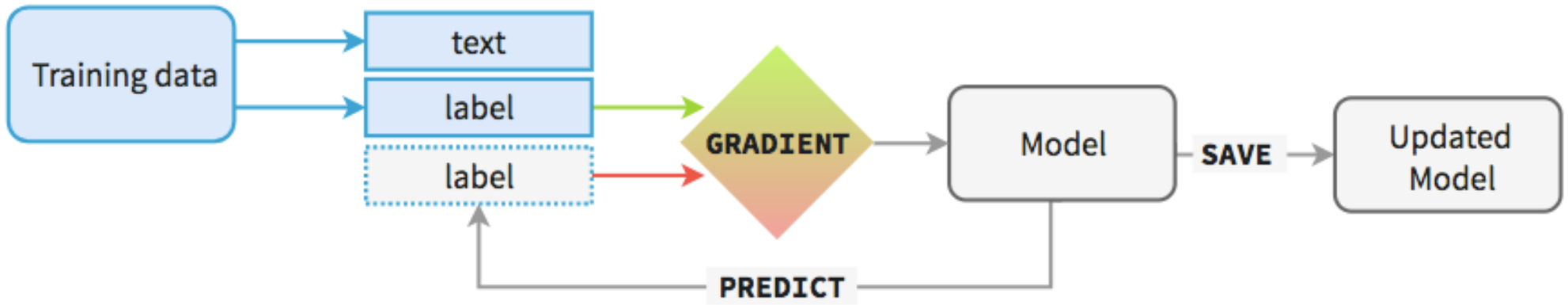
Processing pipelines

```
text = u"Apple is looking at buying U.K. startup"
doc = nlp(text)
```



- **tensorizer** – create feature representation tensor for `Doc`
- **tagger** – assign part-of-speech tags
- **parser** – assign dependency labels
- **ner** – detect and label named entities

Training



- models are **statistical** – every “decision” is a **prediction**
- predictions are based on **labelled examples** the model has seen during training
- model makes prediction of label and receives feedback: **error gradient** of loss function that calculates difference of training example and expected output

Resources

Documentation

spacy.io/docs/usage

Source & issue tracker

github.com/explosion/spacy

spaCy 101 (alpha)

alpha.spacy.io/docs/usage/spacy-101

Thanks!

💥 **Explosion AI**

explosion.ai

📲 **Follow us on Twitter**

@honnibal

@_inesmontani

@explosion_ai