spaCy

Présentation par Morgann Sabatier

Inspirée par :

- Formation spaCy.io : **spaCy 101 : Everything you need to know**

- **La documentation spaCy – spacy.io**

- Tutoriel realpython.org : **Natural Language Processing With spaCy in Python**

- **Traitement Automatique du Langage Naturel en français (TAL / NLP)** par Maël Fabien

Objectifs

01. Comprendre

02. Présenter

03. Nuancer

spaCy, c'est quoi ?

2015
Matthew Honnibal et Ines Montani

Librairie Python **open-source**
Outils flexibles

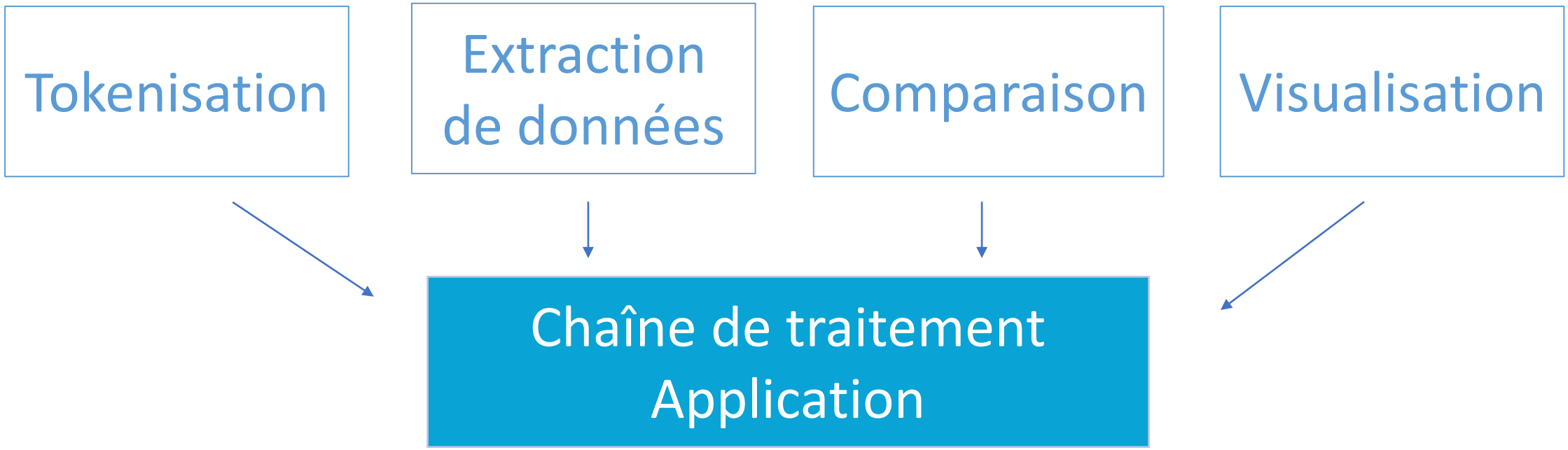
Licence MIT – logiciel libre

65 modèles, 17 langues

Communauté active

Pour quoi faire ?

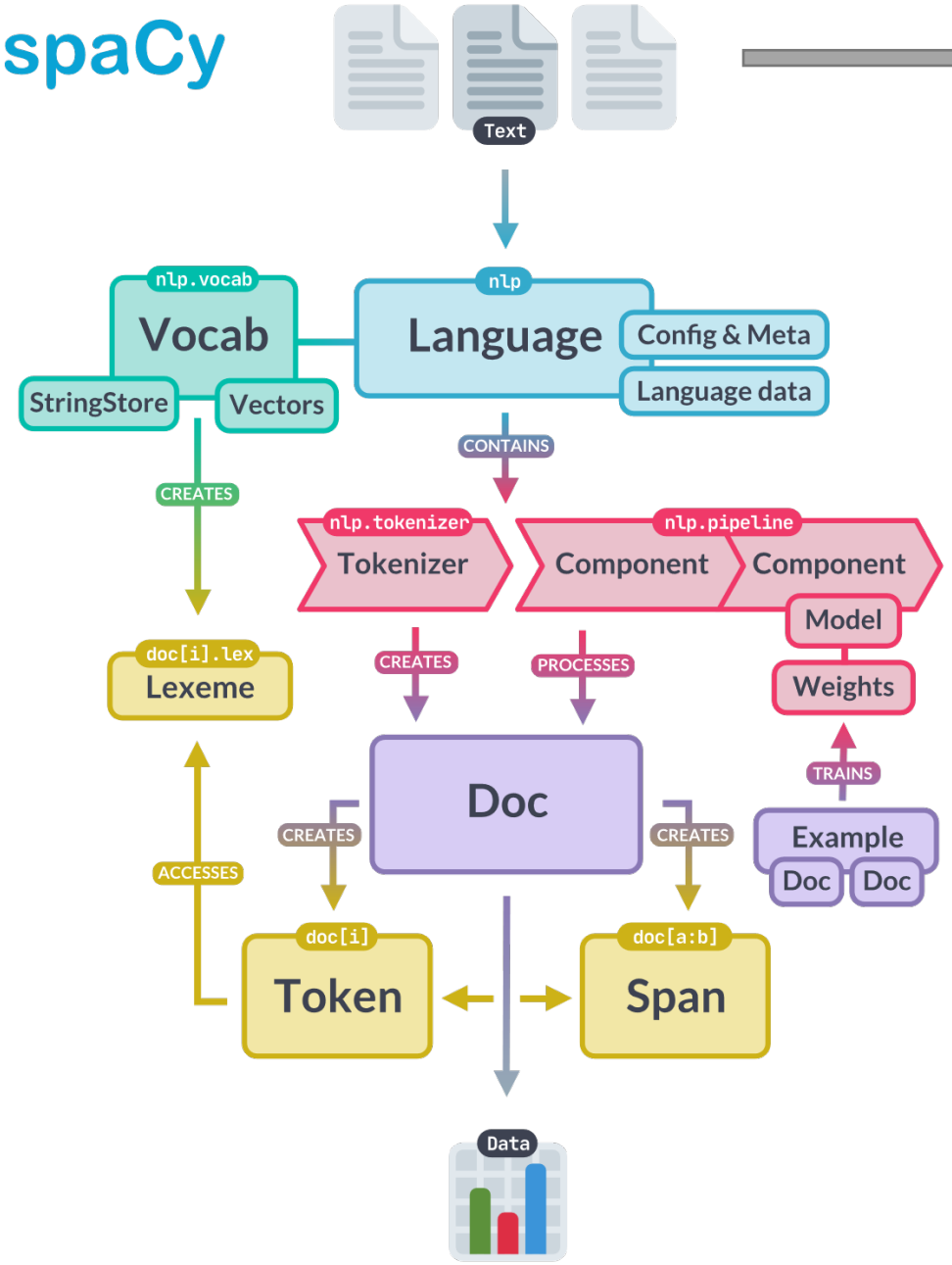
NLTK → Enseignement et recherche
spaCy → production et application



Comment ça marche ?

- Architecture
- Pipeline
- Format de données
- Modèles

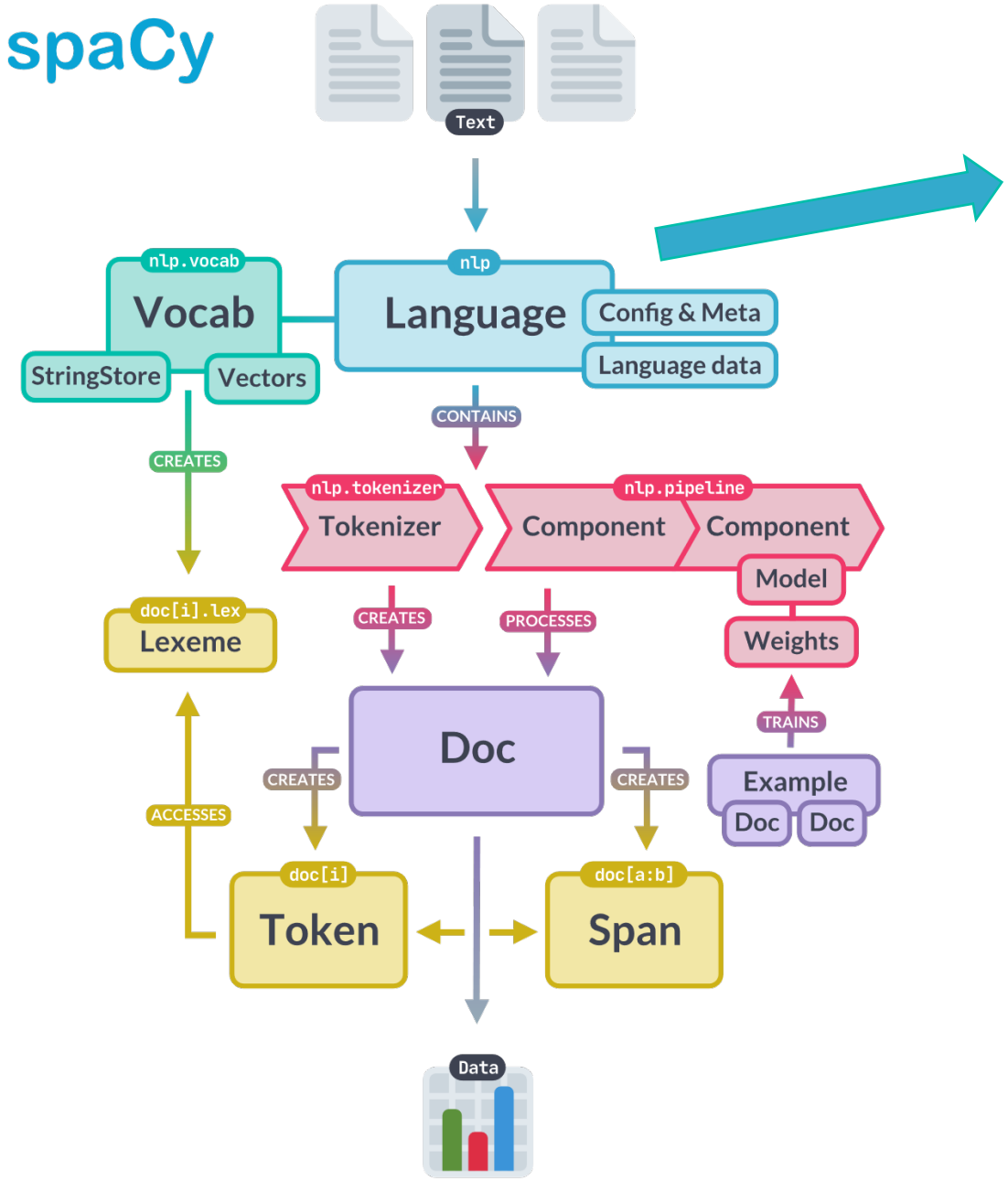
spaCy




```
text = "Morgann explains how spaCy works"
```

- Chaines de caractères
- Documents
- Corpus

Source : https://spacy.io/usage/spacy-101#architecture

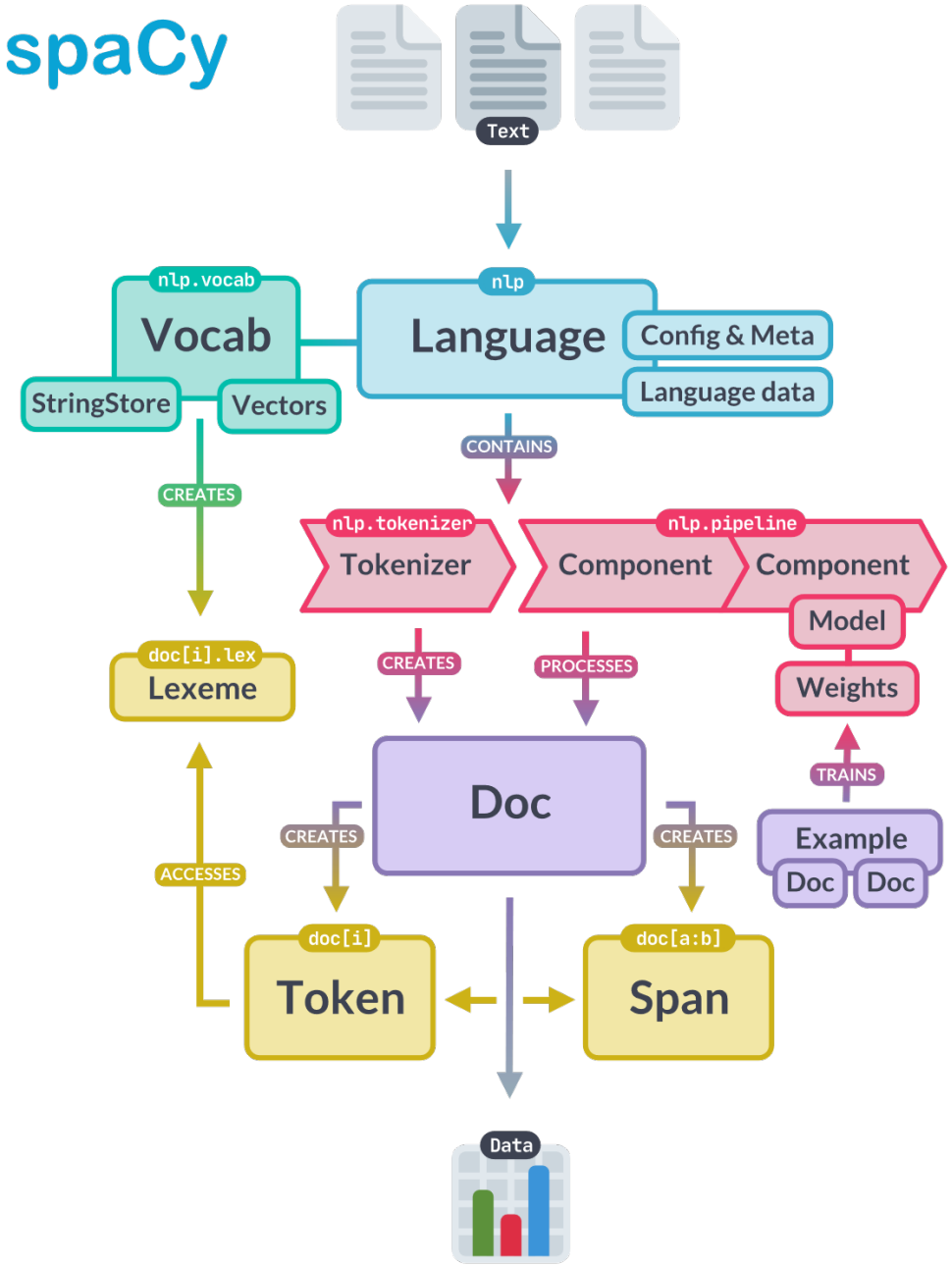



```
text = "Morgann explains how spaCy works"
```

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

- Vocabulaire du modèle
  - Vecteurs + strings
- Lexèmes

Source : https://spacy.io/usage/spacy-101#architecture



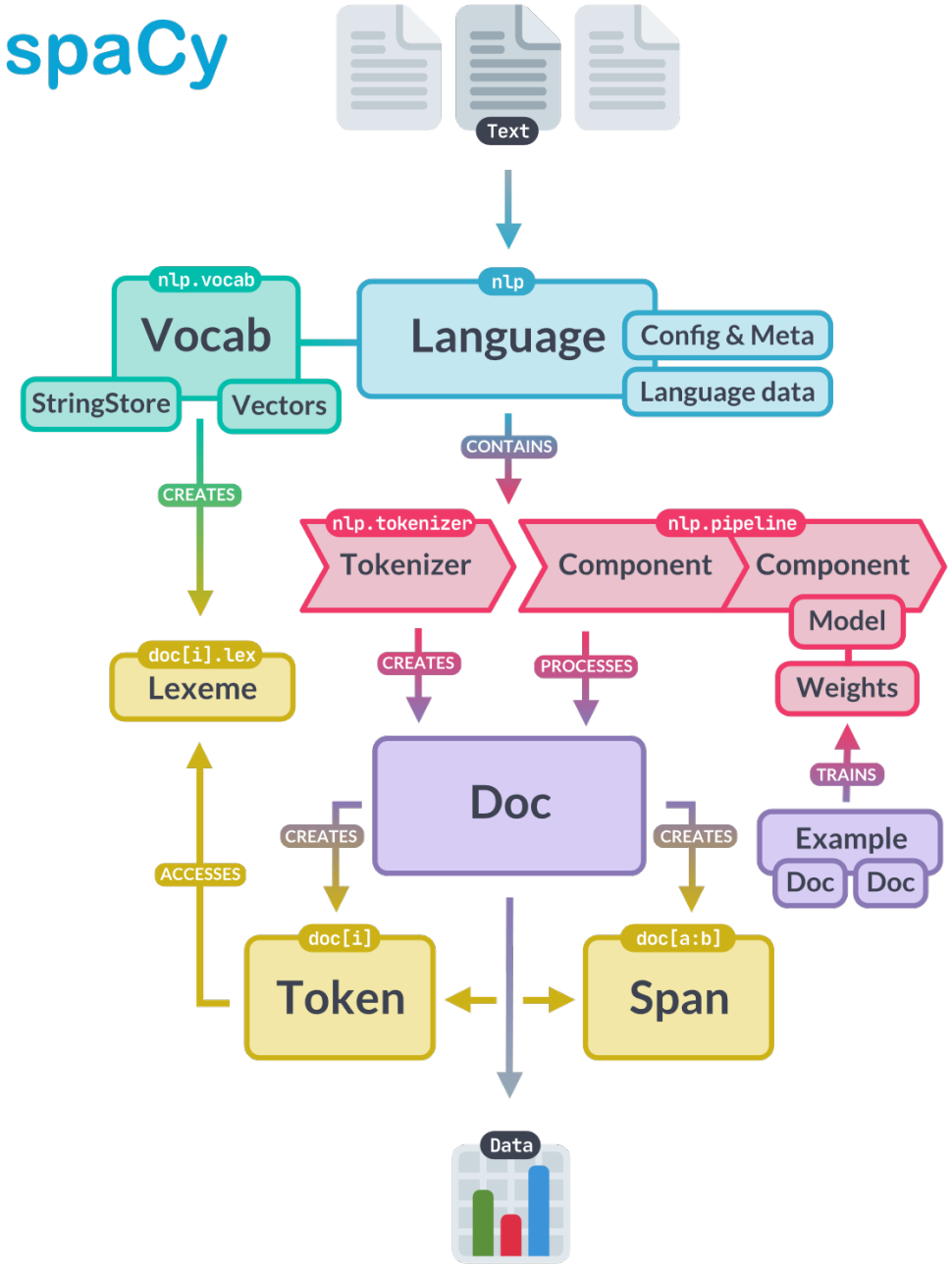

```
text = "Morgann explains how spaCy works"
```

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

```
doc = nlp(text)
```

- Tokenizer
- Parser (dependencies)
- Etiquetage morpho syntaxique
- Reconnaissance entités nommées
- Vos composants

Source : https://spacy.io/usage/spacy-101#architecture




```
text = "Morgann explains how spaCy works"
```

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

```
doc = nlp(text)
```

TOKENISATION

ANNOTATION

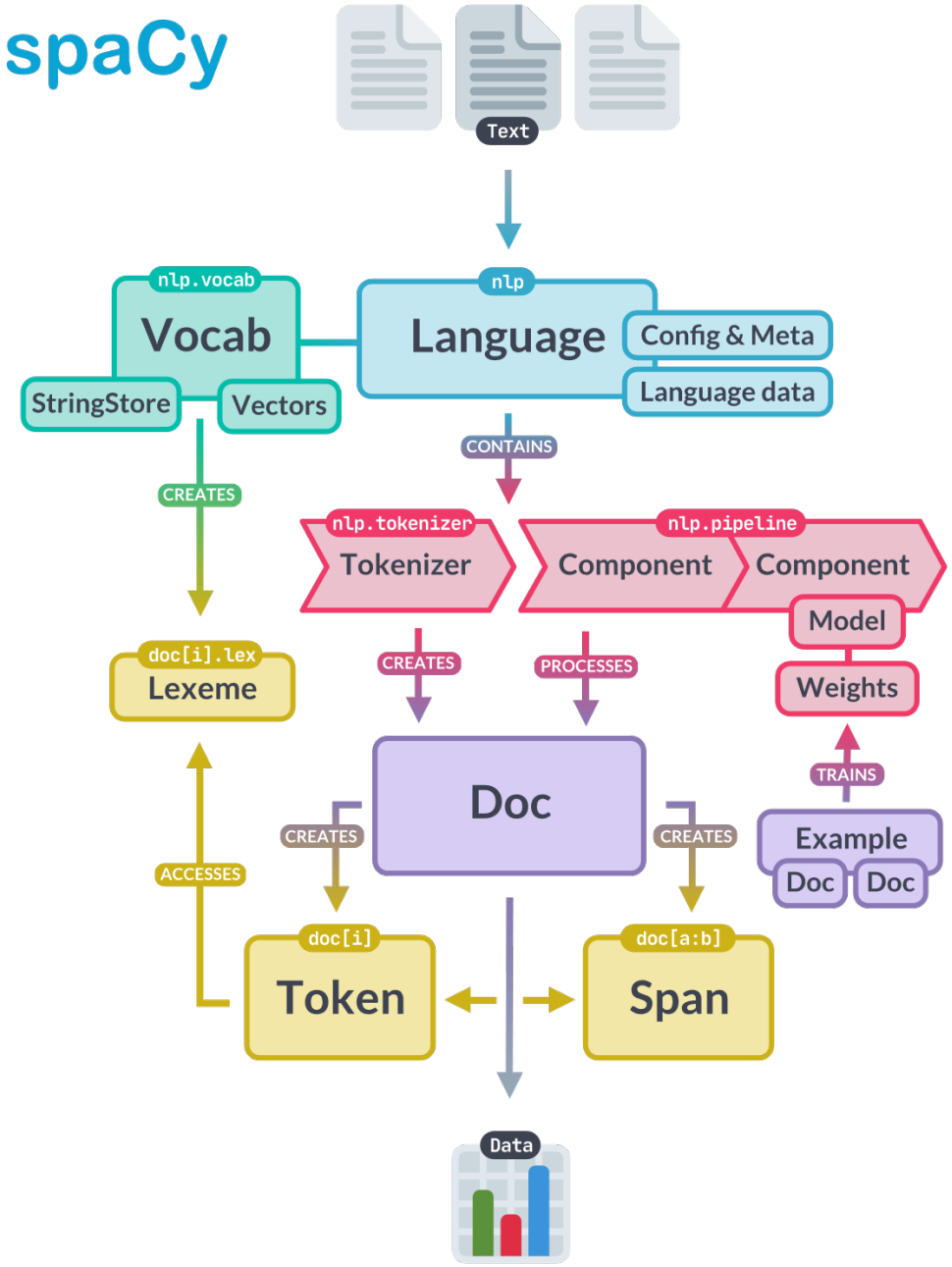
```
Doc[0] = Morgann                    #Token
```

```
Doc[0].pos_ = PROPN                 #Nom propre
```

```
Doc[0].dep_ = nsubj                 #Sujet
```

```
[(ent.text, ent.label_) for ent in doc.ents]
>>> [('Morgann', 'PERSON')]
```

Source : https://spacy.io/usage/spacy-101#architecture



Source : https://spacy.io/usage/spacy-101#architecture

```
text = "Morgann explains how spaCy works"
```

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

```
doc = nlp(text)
```

TOKENISATION

ANNOTATION

```
Doc[0] = Morgann                    #Token
```

```
Doc[0].pos_ = PROPN                 #Nom propre
```

```
Doc[0].dep_ = nsubj                 #Sujet
```

```
[(ent.text, ent.label_) for ent in doc.ents]
>>> [('Morgann', 'PERSON')]
```

ENRICHISSEMENT

```
spacy_ent = Span(doc, 3, 4, label="LIB")
doc.ents = list(doc.ents)+[Spacy_ent]
>>> [('Morgann', 'PERSON'), ('spaCy', 'LIB')]
```

Fonctionnalités

Quelles fonctionnalités ?

- Tokenisation
- Token attributes
- Entités nommées
- Matcher

Tokenisation

spacy.tokens.doc.Doc VS **list** NLTK

NLTK	['d', ''', 'enseignants'], [l'encouragement] ['O.N.U', '.'] ['#', 'France']
spaCy	['d'', 'enseignants'], ['l'', 'encouragement'] ['O.N.U.'] ['#', 'France']

Et pour le mandarin ?

NLTK	['我想自我介绍一下']
spaCy	['我', '想', '自我', '介绍', '一下']

Et les langues agglutinantes ?

Token attributes

- **Extraire** des informations

.lemma_	.tag_	.morph	Spacy.explain(token.tag_)
we	PRP	Number=Plur|Person=1|PronType=Prs	Pronoun, personal

- **Interroger** un type

Token.text	token.is_alpha	Token.is_punct	Token.shape_	Token.is_stop
Name	True	False	Xxxx	True

Stop Words

NLTK, mots vides :

```
il', 'ils', 'je', 'la',
'le', 'les', 'leur',
'lui', 'ma', 'mais’
```

spaCy, des mots vraiment vides ?

```
'différents', 'via',
'nombreuses’, 'néanmoins',
'certains', 'toujours’
```

Limite des modèles

Entités nommées

Reconnaissance d'entités nommées + visualisation

Performant en anglais

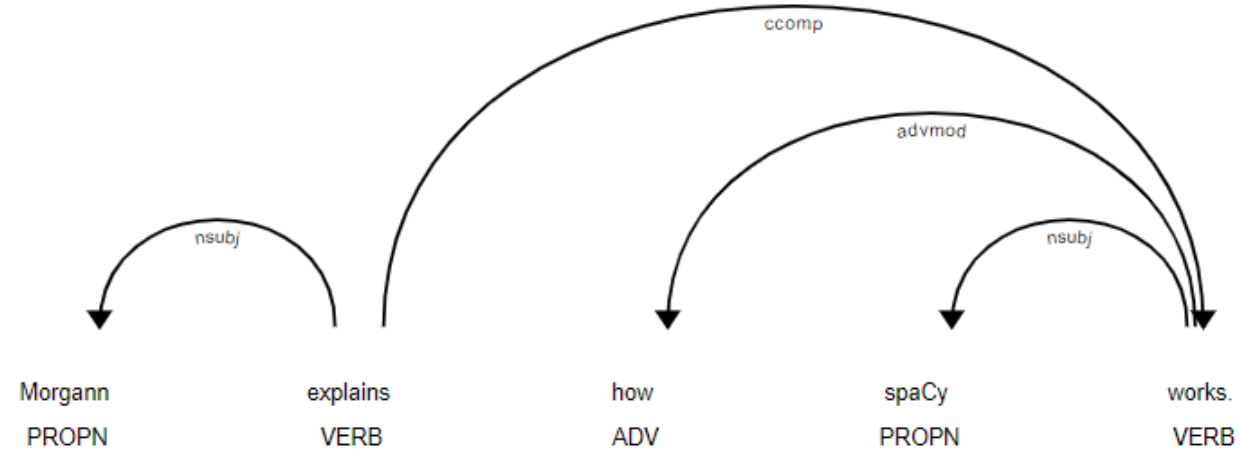
. Marie Curie PERSON , née Maria Sklodowska PERSON , was born in Warsaw GPE on November 7, 1867 DATE

Résultats **dépendants** de l'annotation corpus modèle

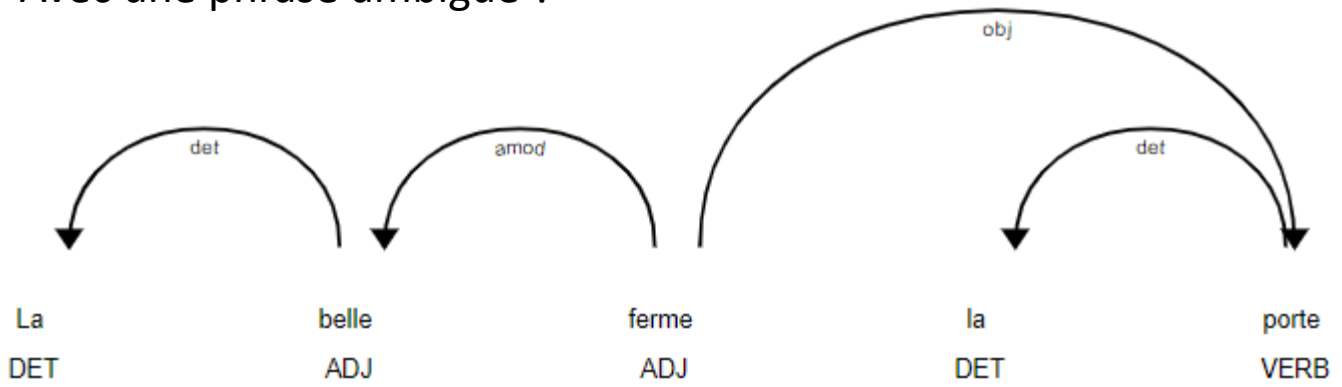
Maria Sklodowska PER naît le 7 novembre 1867 à Varsovie LOC ,

Га́рри Ки́мович Каспа́ров PER (фамилия при рождении Вайнште́йн; род. 13 апреля 1963, Баку LOC , Азербайджанская ССР LOC .

Dependencies

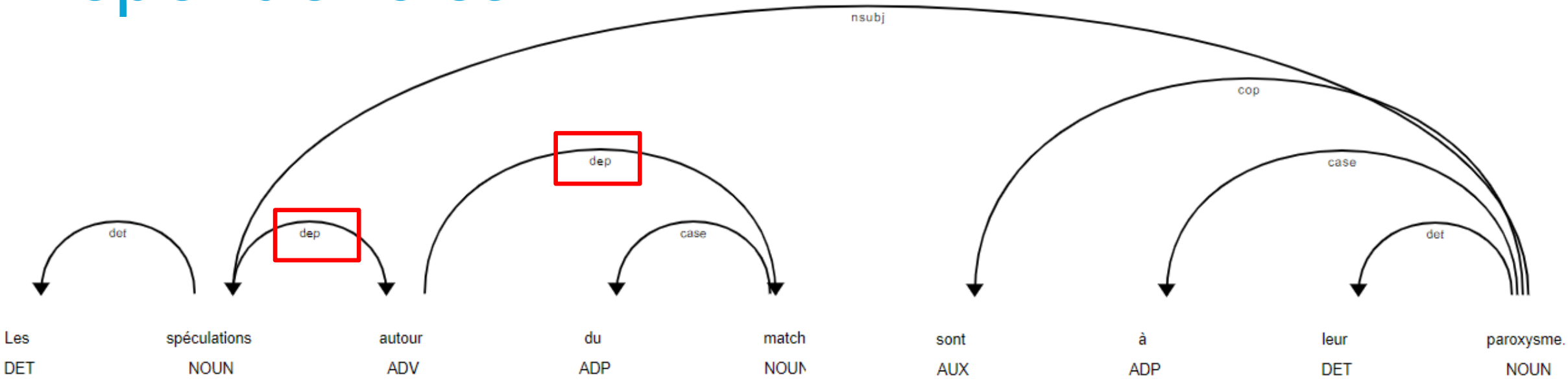


Avec une phrase ambigüe ?

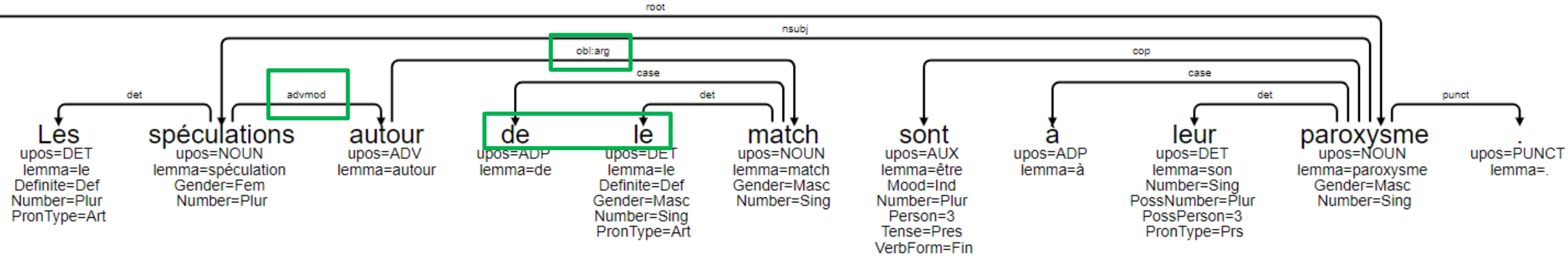



```
displacy.render(doc_fr, style="dep")
spacy.explain("dep") >>> 'adjectival modifier'
```

Dependencies



spaCy visualisation displacy



GREW Match

Matcher


```
{"POS":"ADJ"}, {"POS":"NOUN"}
```

```
Token.pos_
```

```
String ID : ADJ + Nom --
-- Texte match : petit contrat
String ID : ADJ + Nom --
-- Texte match : différents aciers
```

Démo

Bilan

Avantages

+ Simple d’accès
+ Communauté active
+ Performant
+ Customisable
+ Open source et licence libre

Inconvénients

- Modèles hors anglais peu performants
- Pas de langues peu dotées
- Puissance des outils limités
- Stopwords
- Ambiguïté

Merci de votre attention

Je mets à votre disposition

- La présentation en format PDF
- Un notebook avec des exemples simples et en différentes langues de nombreux outils de spaCy
- Le notebook de la démo présentée

Quelques ressources :

- http://datacamp-community-prod.s3.amazonaws.com/29aa28bf-570a-4965-8f54-d6a541ae4e06 - Cheatsheet fonctions
- https://betterprogramming.pub/extract-keywords-using-spacy-in-python-4a8415478fbf - Trouver Keywords pipeline
- https://towardsdatascience.com/named-entity-recognition-ner-using-spacy-nlp-part-4-28da2ece57c6 - NER