Lundi 20 Janvier 2020

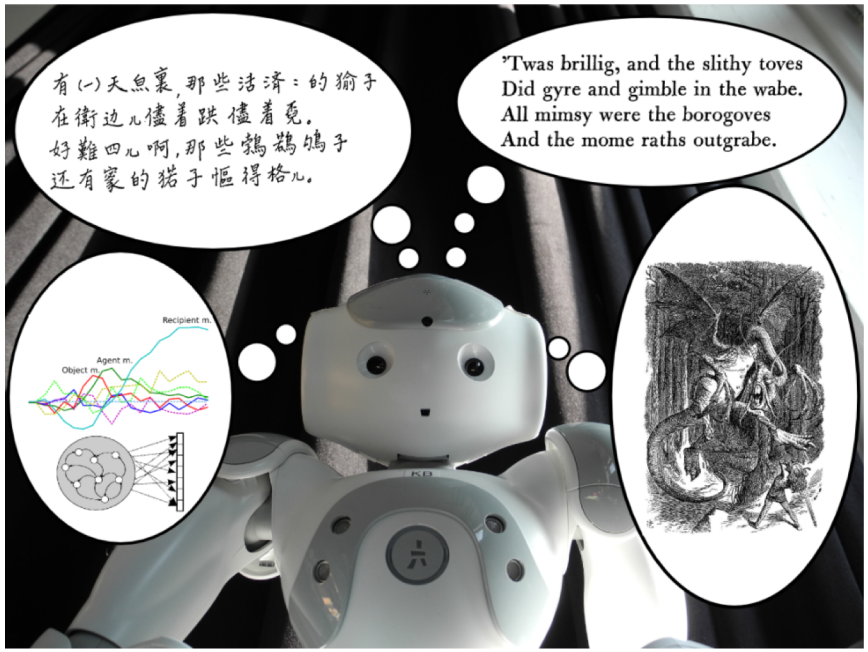
Workshop « *AI 4 Industry* », Bordeaux

Une Introduction au Traitement Automatique du Langage

Xavier HINAUT

Chercheur Inria informatics mathematics dans l'équipe Mnemosyne

Computational Neuroscience Cognition Memory Brain & Body Complex systems



@neuronalX

github.com/neuronalX

www.xavierhinaut.com

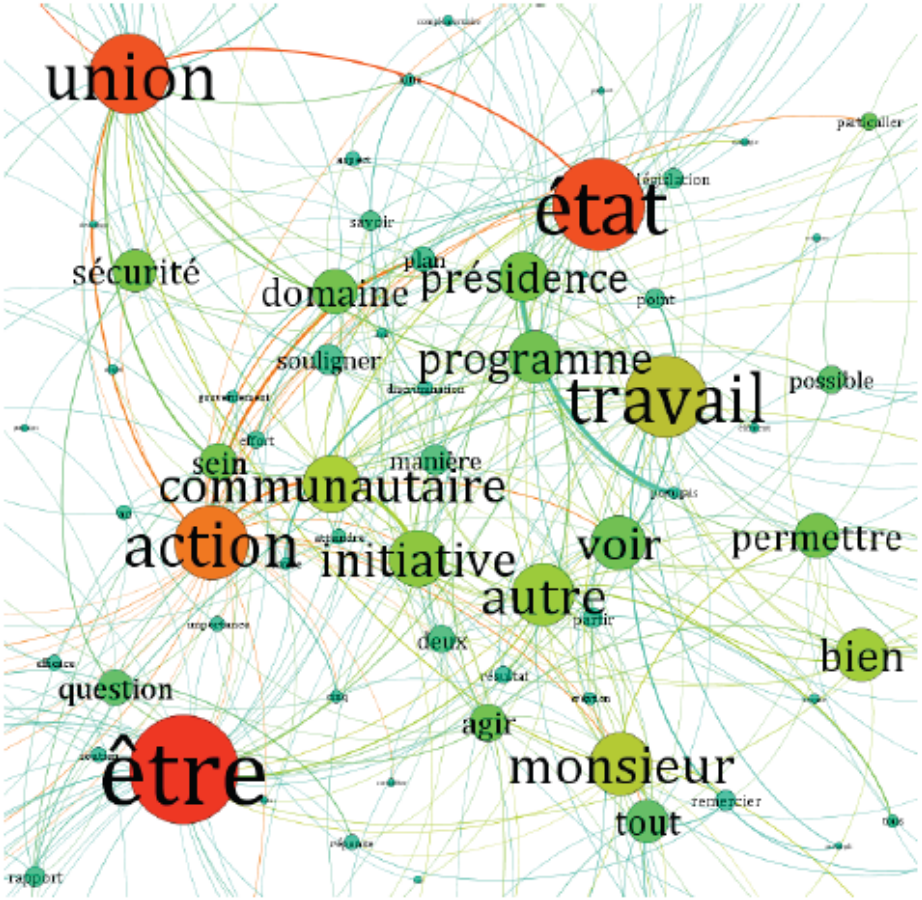
xavier.hinaut@inria.fr

« gavagai »

(CC0) maxpixel.freegreatpicture.com/Pet-Spring-Bunny-Garden-Rabbit-717857

Comment extraire de l'information de mots ?

(sans rien savoir sur la langue ou presque ...)



utiliser la cooccurrence des mots

apparition des mots dans un même contexte

https://journals.openedition.org/corpus/2236

Query

vin

Language

French (Web, 2012)

Attribute

Lemma

SEARCH

	Similarity	Rank
liquoreux	0.860	47233
crémant	0.859	71504
cru	0.824	7199
chardonnay	0.817	43626
Sauternes	0.814	57729
Vin	0.810	17289
eaux-de-vie	0.810	66759
muscat	0.807	39817
Crémant	0.804	86936
cognac	0.794	23657
spiritueux	0.792	22914
Chardonnay	0.782	36410
chasselas	0.780	66228
liqueur	0.780	12554

Query

Bordeaux

Language

French (Web, 2012)

Attribute

Lemma

SEARCH

	Similarity	Rank
Montpellier	0.872	4504
Libourne	0.871	33640
Toulouse	0.863	3296
Nantes	0.857	4251
Lyon	0.828	2173
Angers	0.821	8486
Dijon	0.821	7166
Rennes	0.818	5003
Pessac	0.811	37646
Bordelais	0.809	21351
Talence	0.806	40974
Langon	0.806	57743
Caen	0.788	7205
Gironde	0.784	11262

Query

Bordeaux vin

Language

French (Web, 2012)

Attribute

Lemma

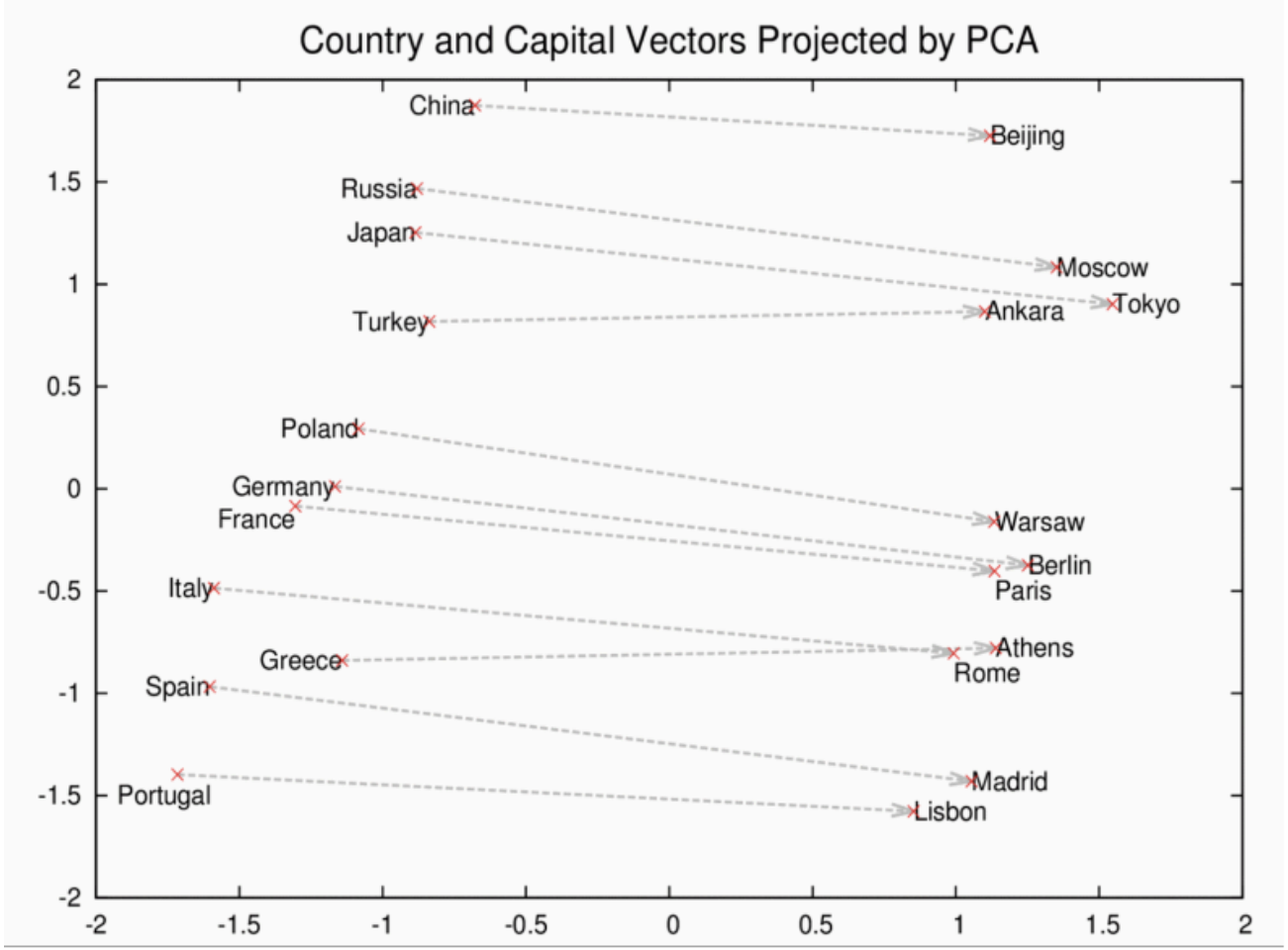
SEARCH

	Similarity	Rank
Sauternes	0.689	57729
Saint-Emilion	0.686	51320
Emilion	0.674	60178
Bordelais	0.673	21351
Chablis	0.665	47174
Vins	0.665	21654
Pomerol	0.659	61497
Pauillac	0.655	57868
Beaujolais	0.654	22375
Vin	0.649	17289
Médoc	0.647	26966
Cognac	0.644	20947
Vouvray	0.641	77442
beaujolais	0.641	41013
Gaillac	0.639	40721

Testez par vous même !

- Sketch Engine:
  - Test word representation:
  - embeddings.sketchengine.co.uk/static/index.html
- fastText
  - github.com/facebookresearch/fastText

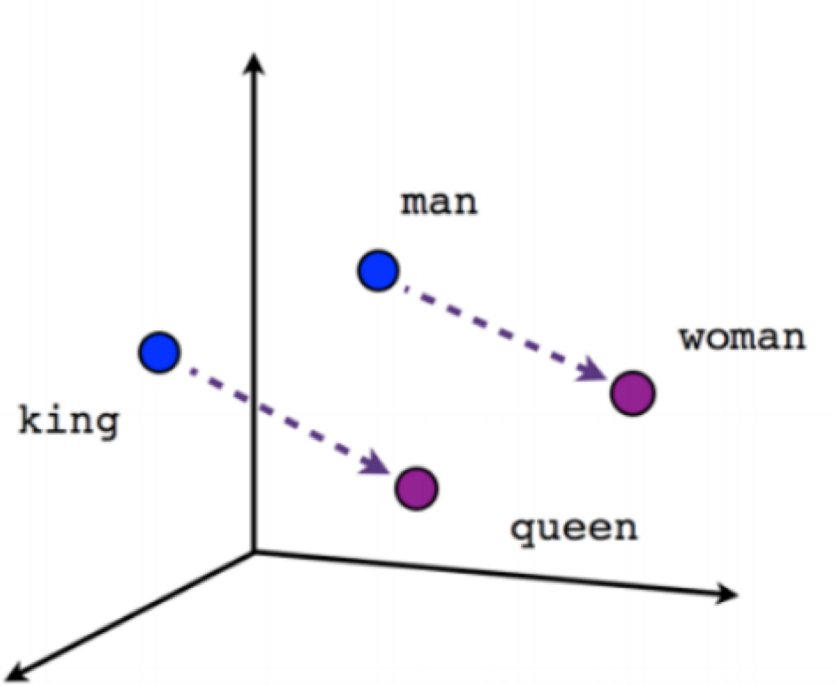
Similarités avec *word2vec*



Rome - Italy = Beijing - China

deeplearning4j.org/word2vec

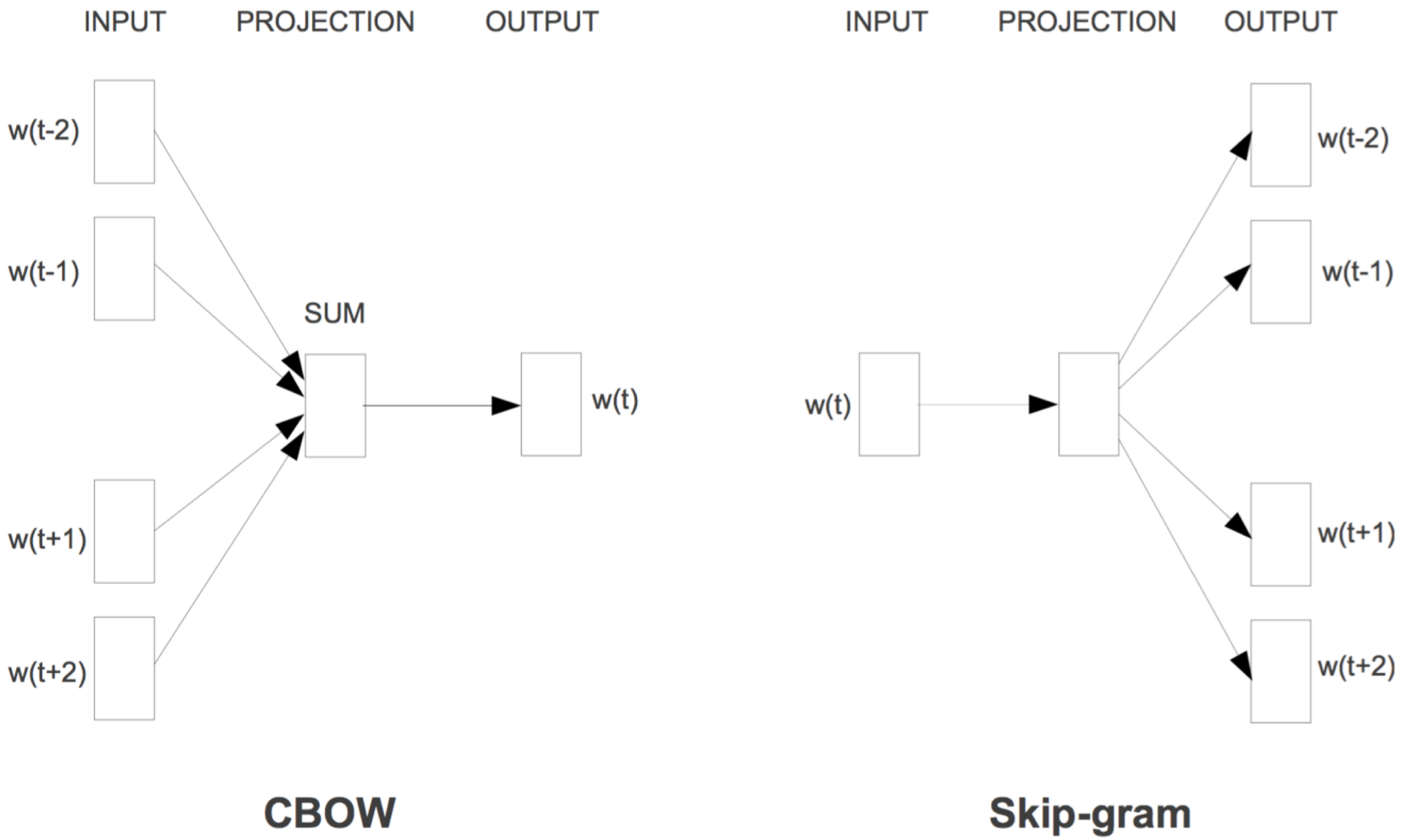
Combinaisons de vecteurs de mots



roi - homme + femme = reine

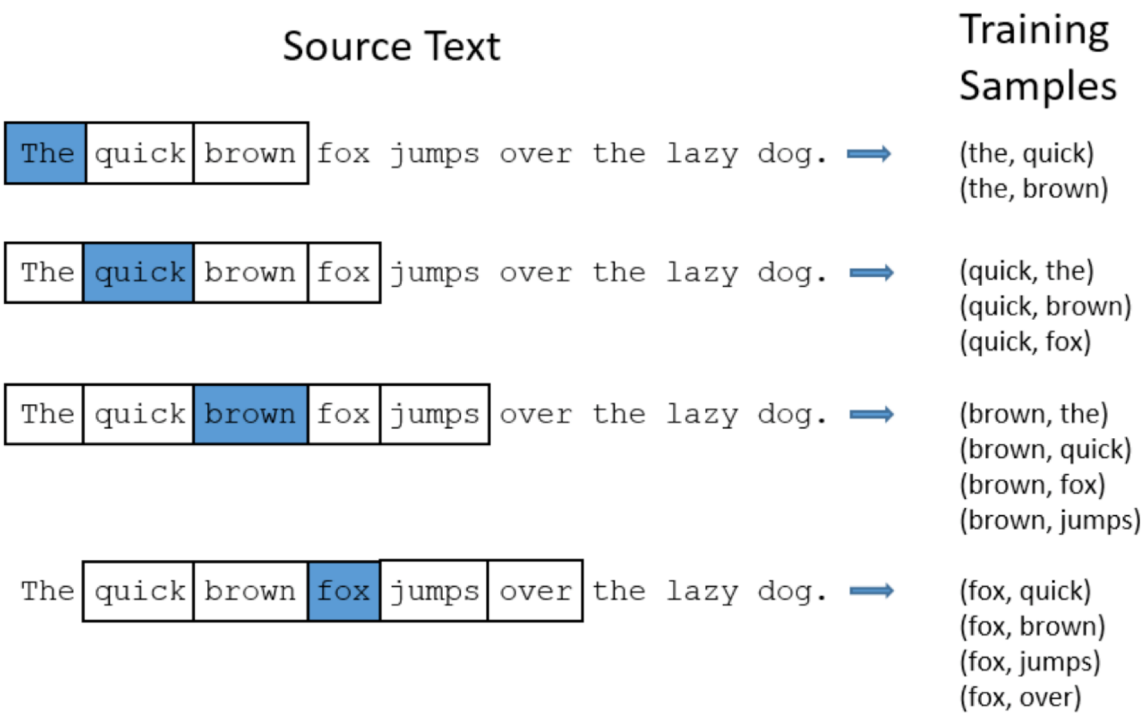
http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

word2vec



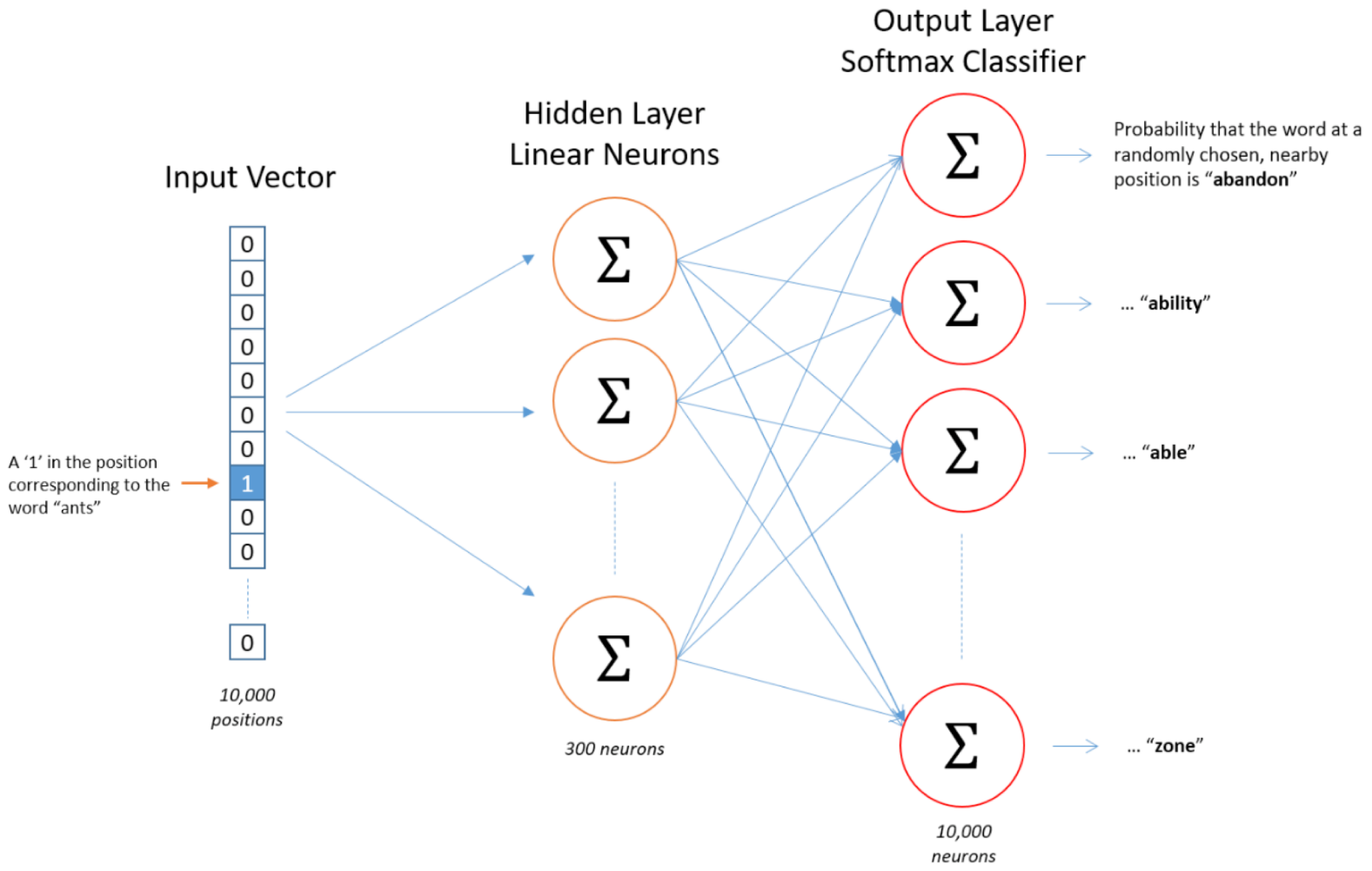
Mikolov et al. 2013

Word2vec - skip-gram



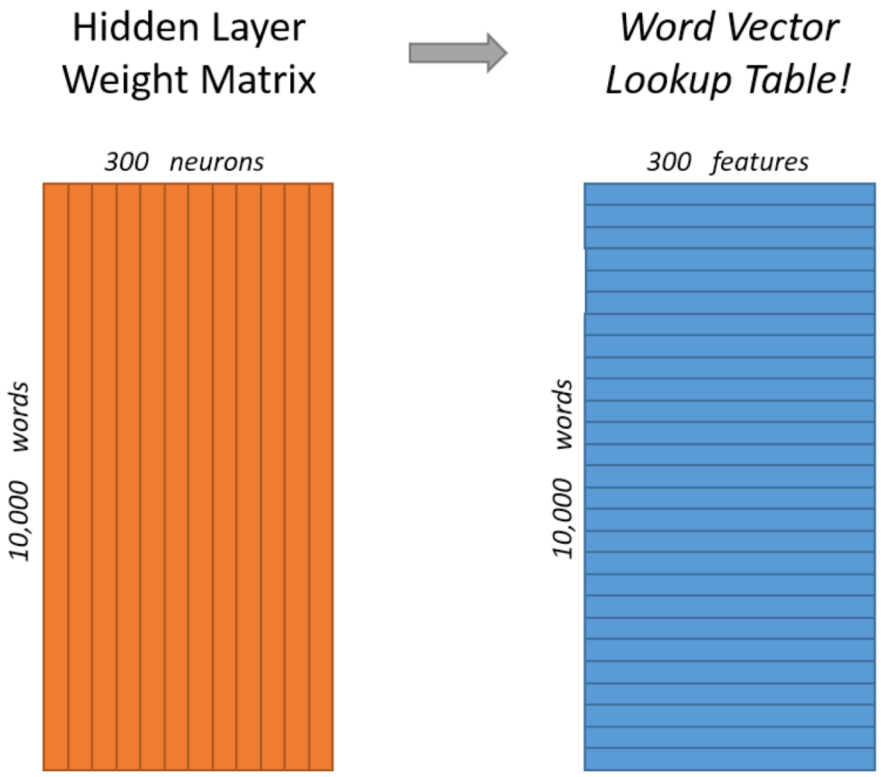
http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Word2vec - skip-gram



http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

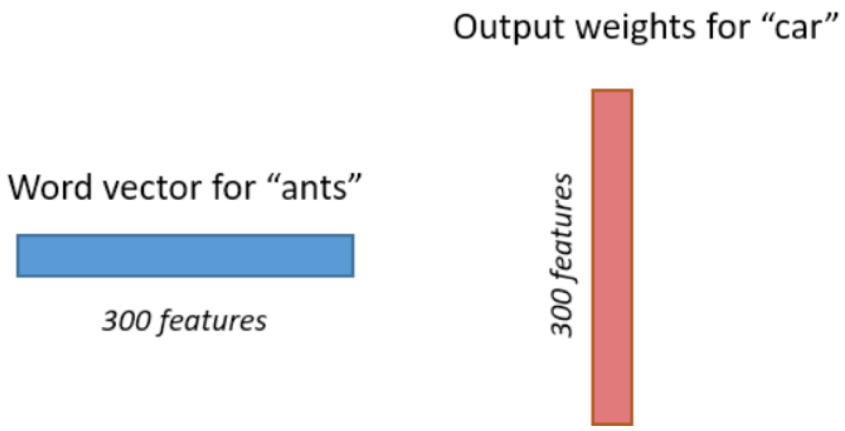
Word2vec - skip-gram



http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

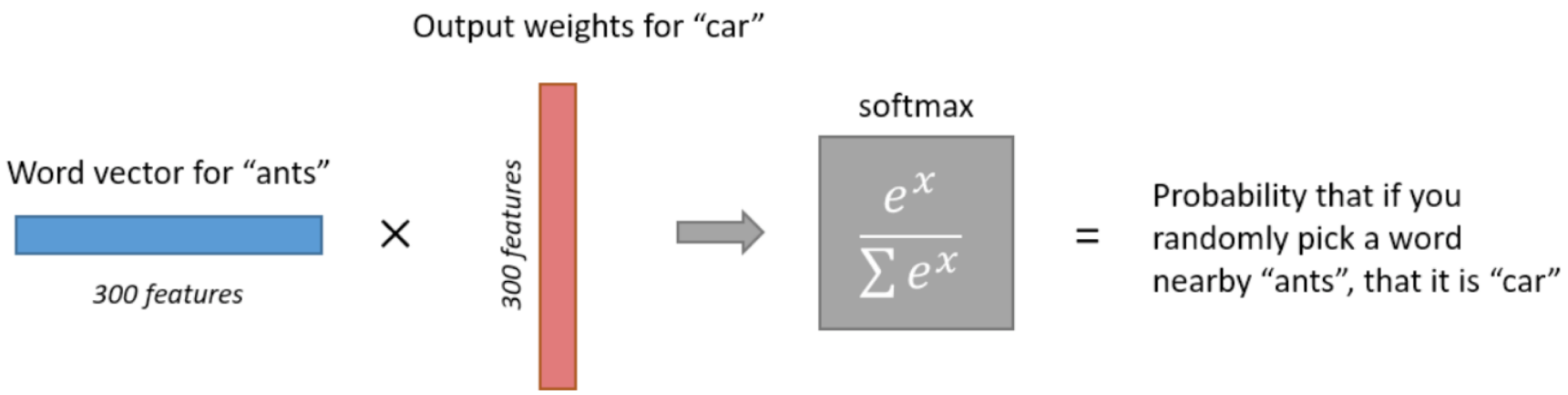
Calcul de similarité

Comment savoir si deux mots apparaissent dans le même contexte ?



http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Calcul de similarité



http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Calcul de similarité avec spaCy

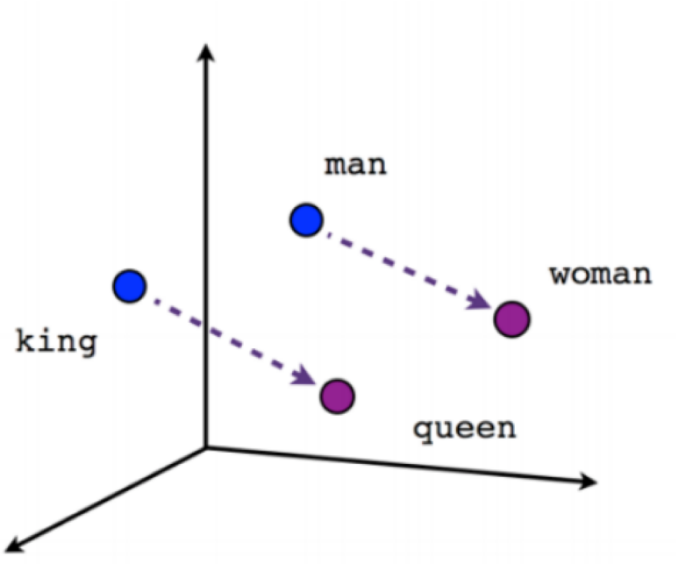
```
# Load a larger model with vectors
nlp = spacy.load('en_core_web_md')

# Compare two documents
doc1 = nlp("I like fast food")
doc2 = nlp("I like pizza")
print(doc1.similarity(doc2))
```

```
0.8627204117787385
```

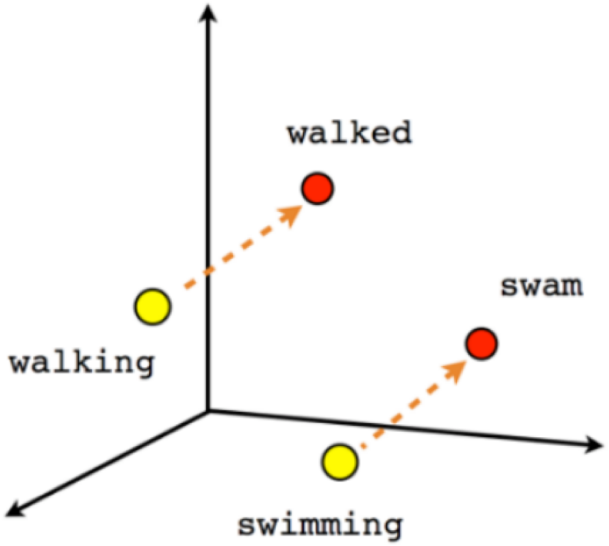
```
# Compare two tokens
doc = nlp("I like pizza and pasta")
token1 = doc[2]
token2 = doc[4]
print(token1.similarity(token2))
```

```
0.7369546
```

Word2vec



Male-Female



Verb tense

http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

skip-gram: negative sampling

- Problème avec les mots les plus courants:
  - "stop words"
  - *le, la, il, de, du, ce, ...*
  - *inclus les mots de fonction (déterminant, préposition, ...)*

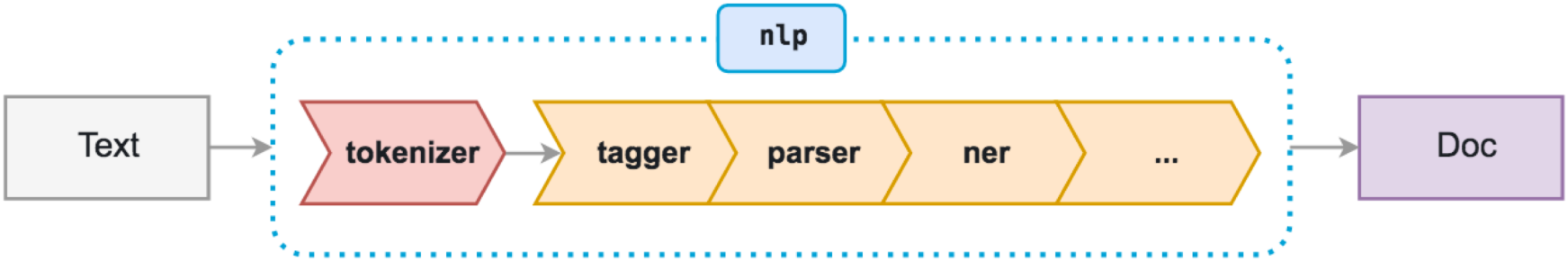
skip-gram: negative sampling

- Problème avec les mots les plus courants:
  - "stop words"
  - *le, la, il, de, du, ce, ...*
  - *inclus les mots de fonction (déterminant, préposition, ...)*

- Adaptation de la méthode initiale pour sous-échantillonner les *stop words*

  ... en pratique cela veut dire qu'il faut faire du *prétraitement*

Aperçu des traitements dans spaCy



www.spacy.io

Tokenization

- Mettre sous forme de *tokens* (jetons, segments, ...)
  - segmenter une phrase (ou un texte) en une liste de mots, ponctuation, ...

```
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("Apple is looking at buying U.K. startup for $1 billion")
for token in doc:
    print(token.text)
```

0	1	2	3	4	5	6	7	8	9	10
Apple	is	looking	at	buying	U.K.	startup	for	$	1	billion

Part-of-Speech (POS) tagging

- *tagguer* (“étiqueter”) les mots
  - étiquetage morpho-syntaxique

- associer aux mots d'un texte les informations grammaticales correspondantes
  - nom commun, verbe, adjectif, adverbe, nom propre, déterminant, ...

TEXT	POS
Apple	PROPN
is	VERB
looking	VERB
at	ADP
buying	VERB
U.K.	PROPN
startup	NOUN
for	ADP
$	SYM
1	NUM
billion	NUM

Lemmatizer

- Lemmatisation (en Français)

- faire correspondre les mots à leur forme “canonique”, sans suffixe:
  - lemme “petit”
    - petit, petite, petits, petites
  - lemme “aimer”
    - aimé, aimée, aimant,

TEXT	LEMMA	POS
Apple	apple	PROPN
is	be	VERB
looking	look	VERB
at	at	ADP
buying	buy	VERB
U.K.	u.k.	PROPN
startup	startup	NOUN
for	for	ADP
$	$	SYM
1	1	NUM
billion	billion	NUM

Recherche d’expressions avec des lemmes

```
pattern = [
    {'IS_DIGIT': True},
    {'LOWER': 'fifa'},
    {'LOWER': 'world'},
    {'LOWER': 'cup'},
    {'IS_PUNCT': True}
]
```

```
doc = nlp("2018 FIFA World Cup: France won!")
```

```
2018 FIFA World Cup:
```

Recherche d’expressions avec des lemmes

```
pattern = [
    {'LEMMA': 'love', 'POS': 'VERB'},
    {'POS': 'NOUN'}
]
```

```
doc = nlp("I loved dogs but now I love cats more.")
```

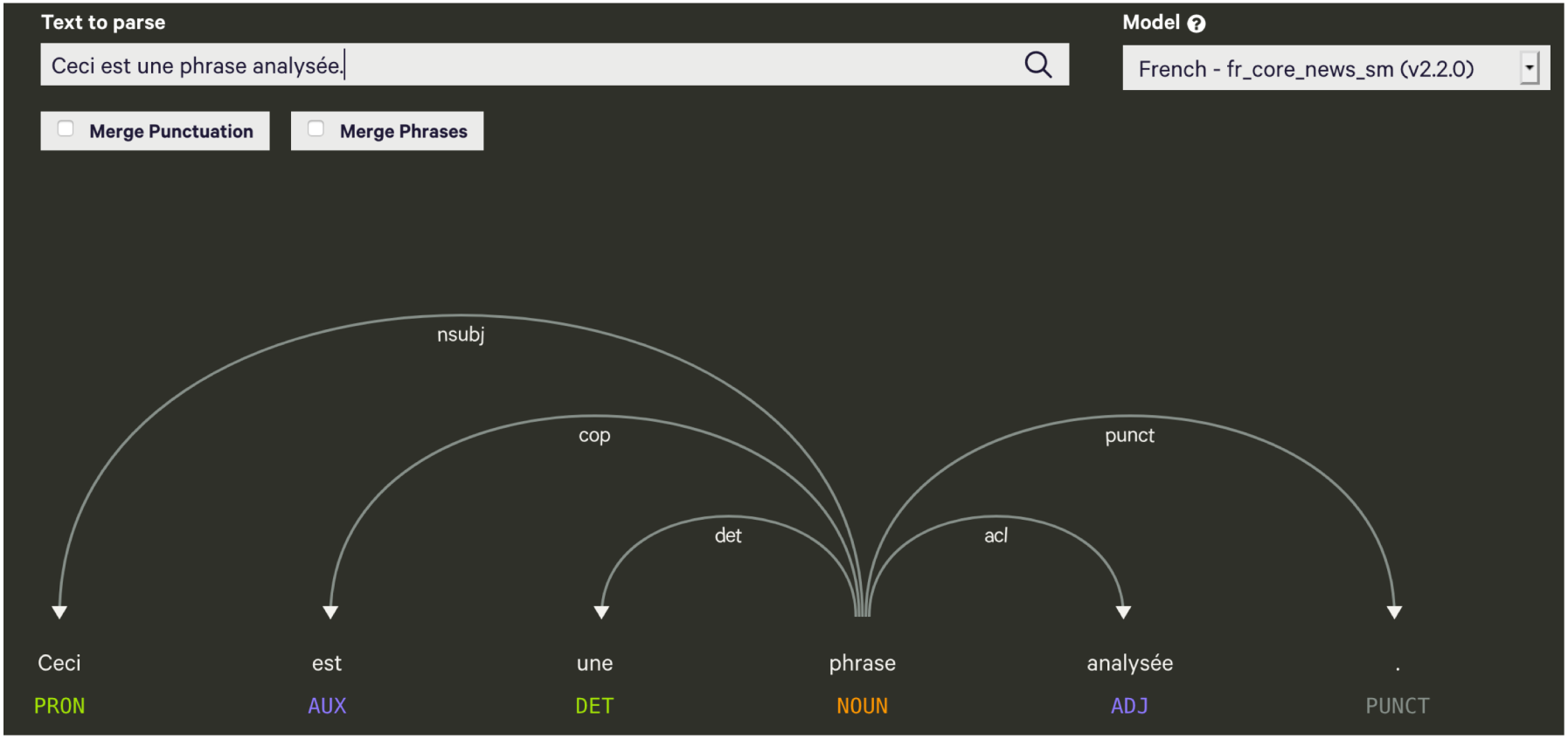
```
loved dogs
love cats
```

Named Entity Recognition

- Reconnaissance d'entités nommées
  - rechercher les mots (ou groupe de mots) catégorisables en classes telles que noms de personnes, noms d'organisations, noms de lieux, quantités, distances, valeurs, dates, etc.

TYPE	DESCRIPTION
PERSON	People, including fictional.
NORP	Nationalities or religious or political groups.
FAC	Buildings, airports, highways, bridges, etc.
ORG	Companies, agencies, institutions, etc.

Analyse syntaxique : arbre de dépendances

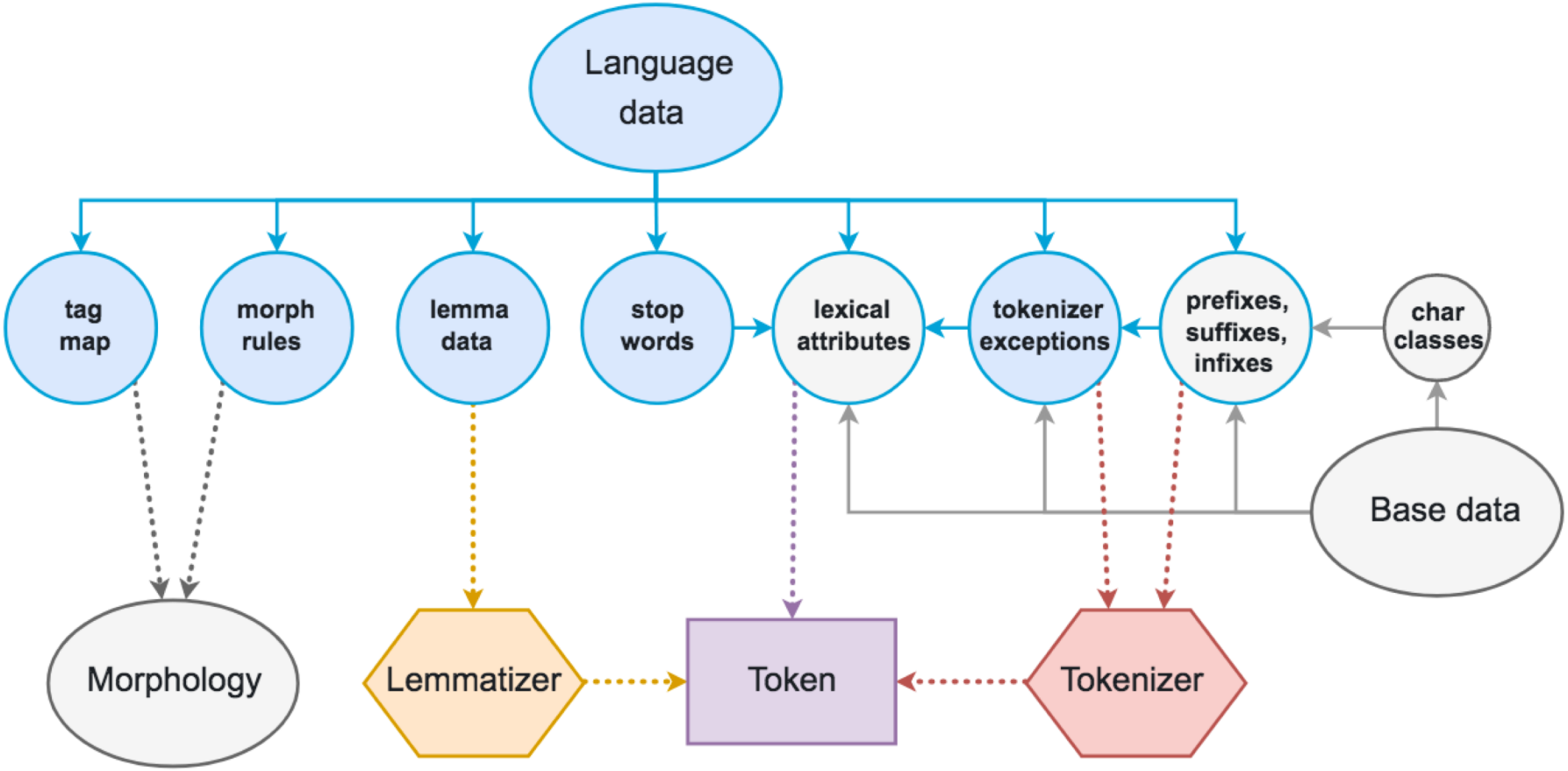


displaCy dependency visualizer

```
import spacy
nlp = spacy.load('en_core_web_sm')
doc = nlp(u"displaCy uses JavaScript, SVG and CSS.")
spacy.displacy.serve(doc, style='dep')
```

www.explosion.ai

Aperçu des traitements dans *spaCy*



Tutoriel officiel de spaCy

course.spacy.io

ADVANCED NLP with spaCy

Chapter 1: Finding words, phrases, names and concepts

This chapter will introduce you to the basics of text processing with spaCy. You'll learn about the data structures, how to work with statistical models, and how to use them to predict linguistic features in your text.

Chapter 2: Large-scale data analysis with spaCy

In this chapter, you'll use your new skills to extract specific information from large volumes of text. You''ll learn how to make the most of spaCy's data structures, and how to effectively combine statistical and rule-based approaches for text analysis.

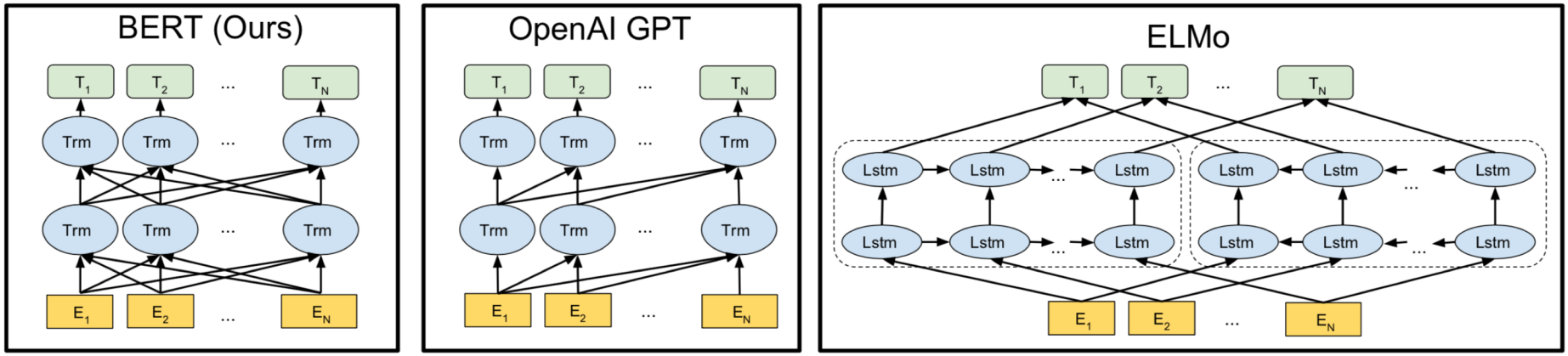
Chapter 3: Processing Pipelines

This chapter will show you to everything you need to know about spaCy's processing pipeline. You'll learn what goes on under the hood when you process a text, how to write your own components and add them to the pipeline, and how to use custom attributes to add your own meta data to the documents, spans and tokens.

Chapter 4: Training a neural network model

In this chapter, you'll learn how to update spaCy's statistical models to customize them for your use case – for example, to predict a new entity type in online comments. You'll write your own training loop from scratch, and understand the basics of how training works, along with tips and tricks that can make your custom NLP projects more successful.

Evolution très rapide des modèles

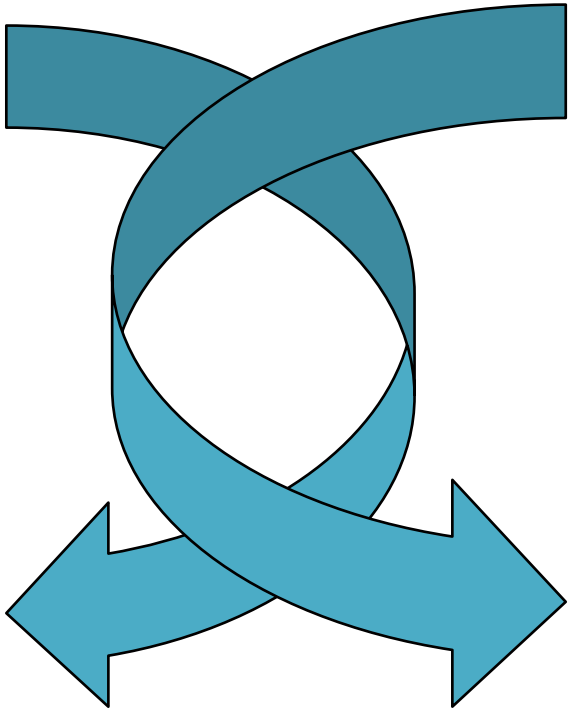


- BERT (Delvin et al, 2018, Google AI Language)
  - architecture complexe
  - améliore des problèmes de *word2vec* ou d'autres représentations
    - polysémie
    - ...

Delvin et al. 2018 arXiv

Modéliser l'apprentissage du langage

iCub



Incarnation (ou Ancrage) du language

Acquisition du langage

Vers la modélisation « in-robota » de l'acquisition du langage

iCub

Ancrage du language

01001100…

"pipe"

Vers la modélisation « in-robota » de l'acquisition du langage

iCub

Ancrage du language

01001100…

10100110…

"pipe"

Vers la modélisation « in-robota » de l'acquisition du langage

Qu'est-ce que la "souplesse" ?

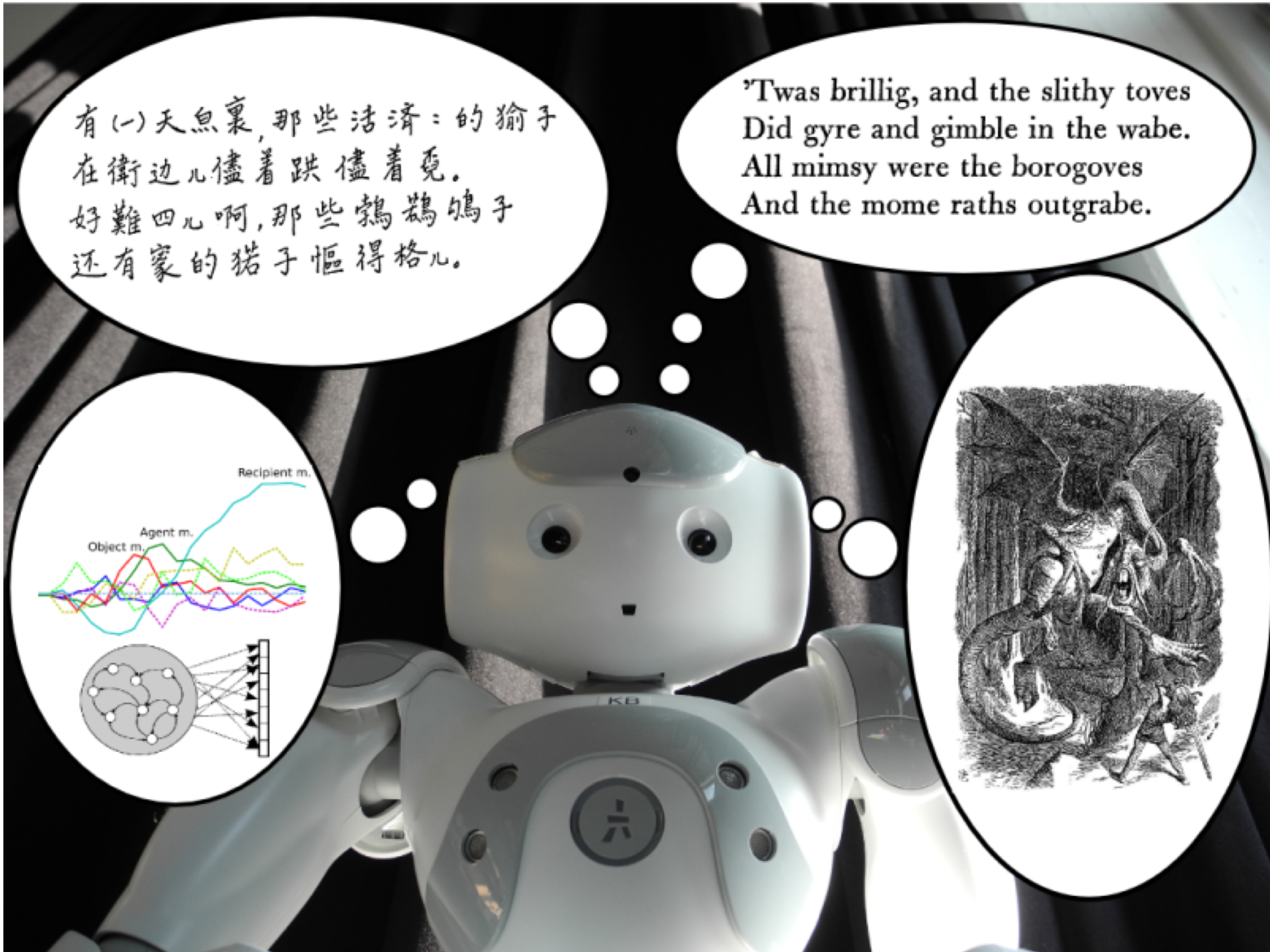
Ancrage du language

Vers la modélisation « in-robota » de l'acquisition du langage

Qu'est-ce que la "souplesse" ?

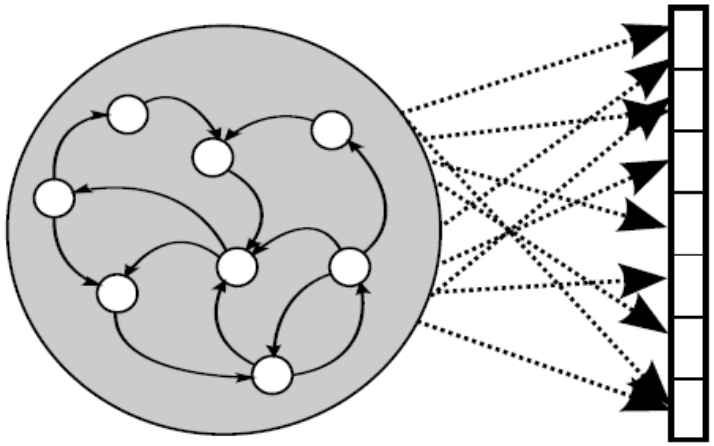
Ancrage du language

Image: Kevin O'Regan (Why red doesn't sound like a bell?)

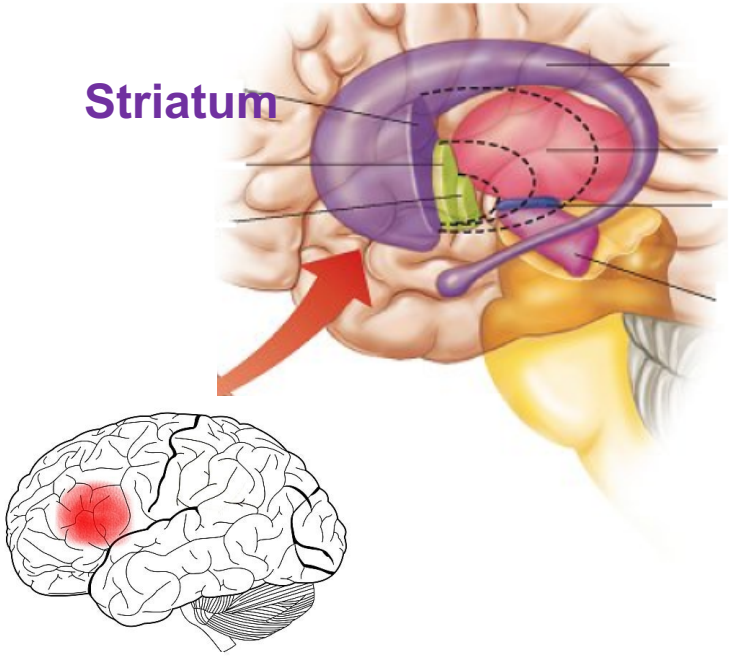


'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe.
All mimsy were the borogoves
And the mome raths outgrabe.
Recipient m.
Agent m.
Object m.

Modéliser l’aire de Broca avec un reservoir

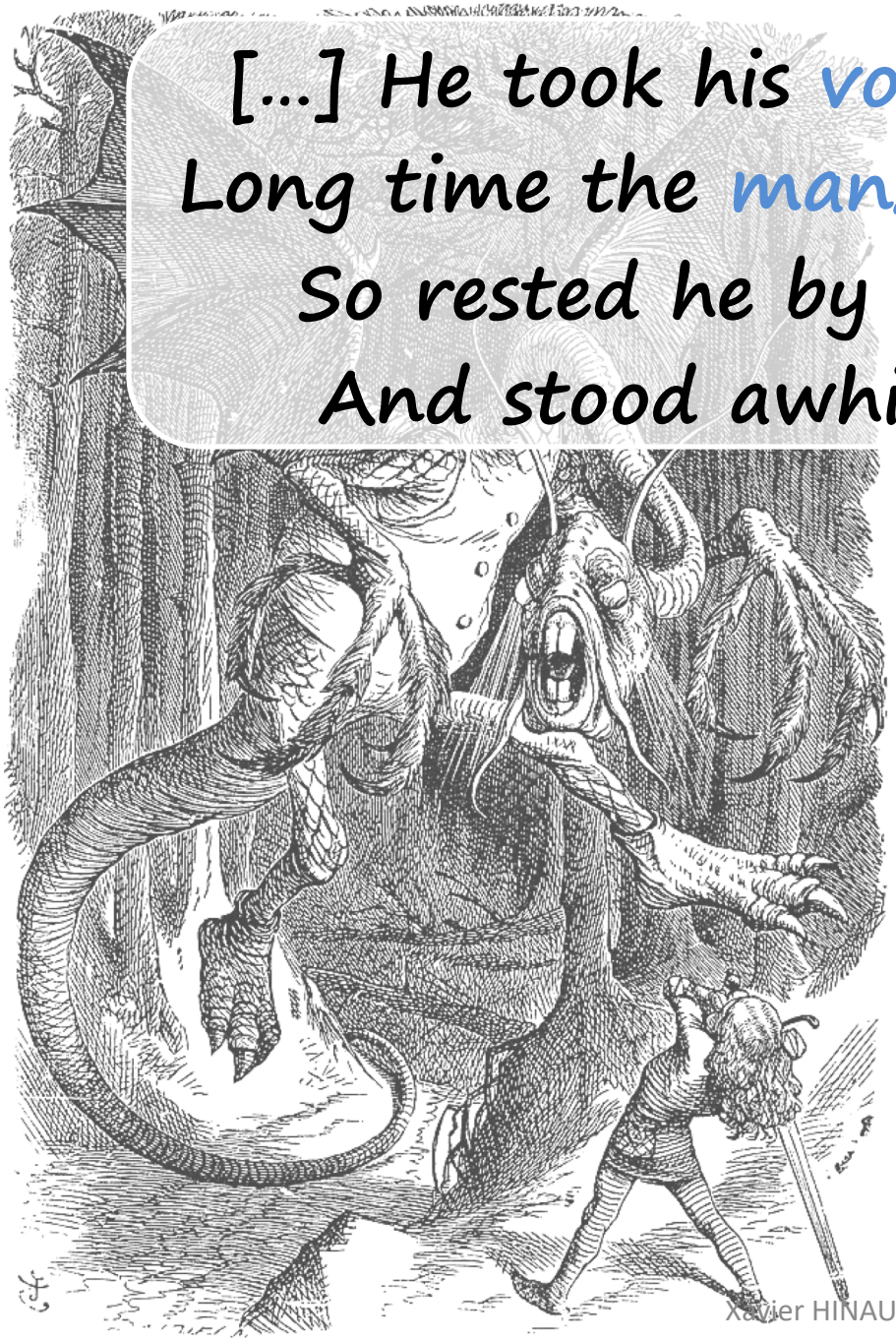


Broca Striatum



[…] He took his vorpal sword in hand:
Long time the manxome foe he sought—
So rested he by the Tumtum tree,
And stood awhile in thought. […]

[…] He took his vorpal sword in hand:
Long time the manxome foe he sought—
So rested he by the Tumtum tree,
And stood awhile in thought. […]



Jabberwocky, Lewis Carroll
Through the Looking-Glass,
and What Alice Found There (1871)

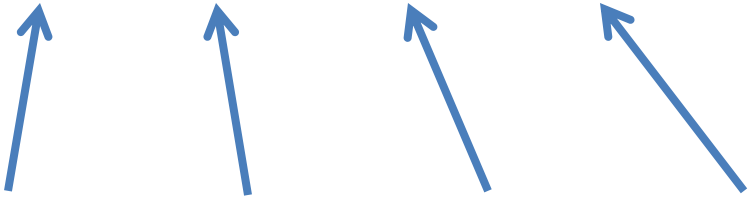
Comprendre une phrase par l'analyse de la syntaxe

- Version minimaliste du problème
- Répondre à la question :
  - « Qui fait quoi à qui ? »

- Qui: agent
- Fait: verbe
- Quoi: objet
- A qui: receveur

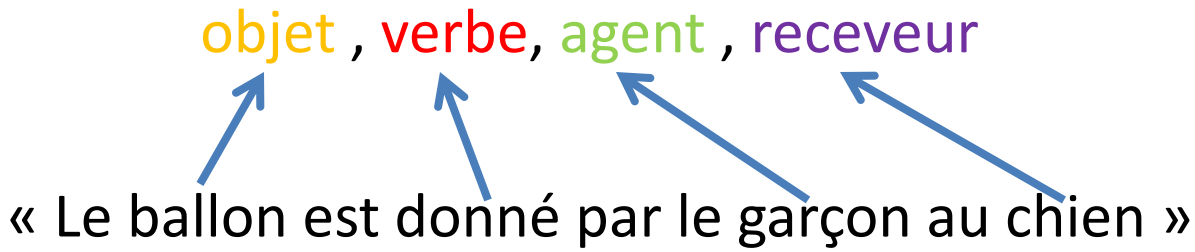
Lier les mots à leur rôle

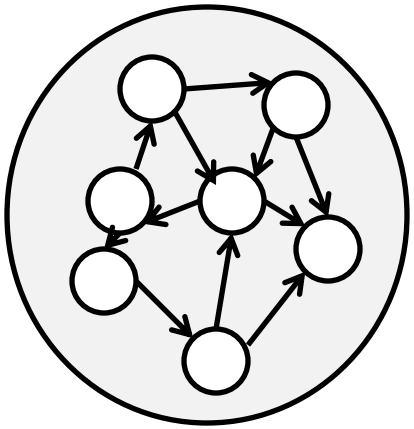
agent , verbe, objet , receveur



« Le garçon a donné le ballon au chien »

Lier les mots à leur rôle





Par

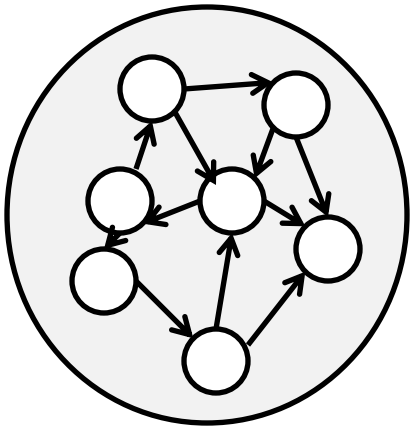
Est

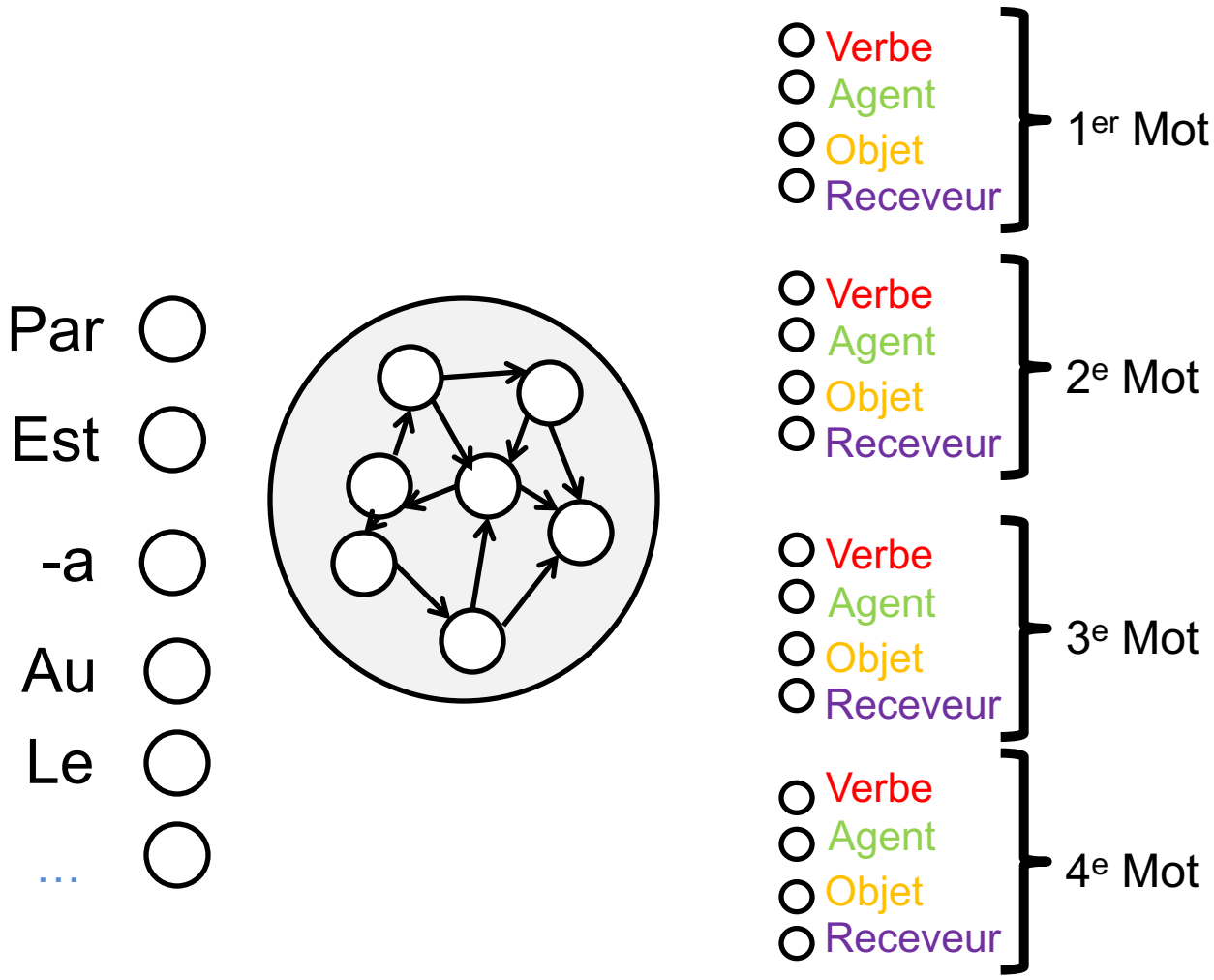
-a

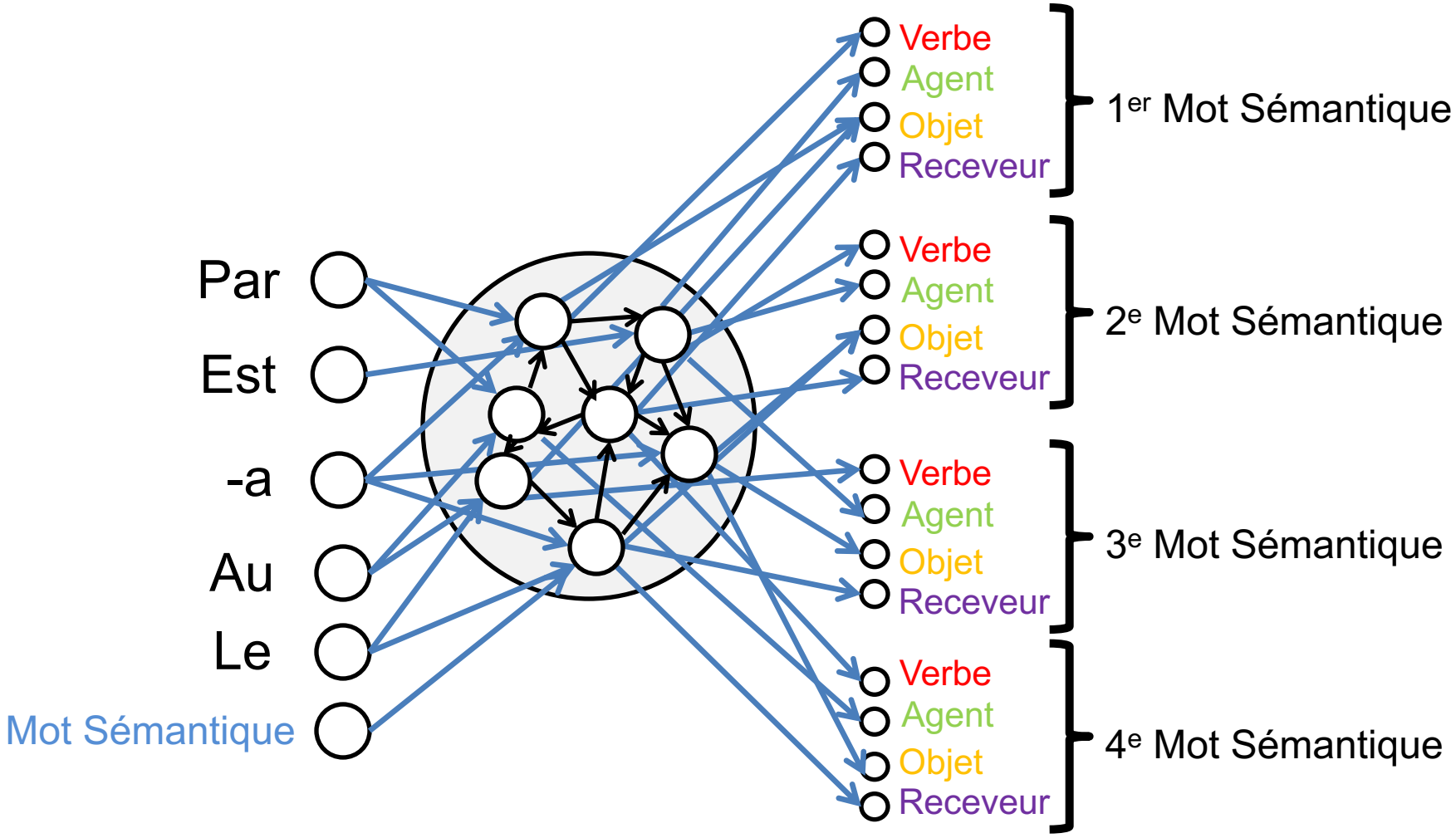
Au

Le

...

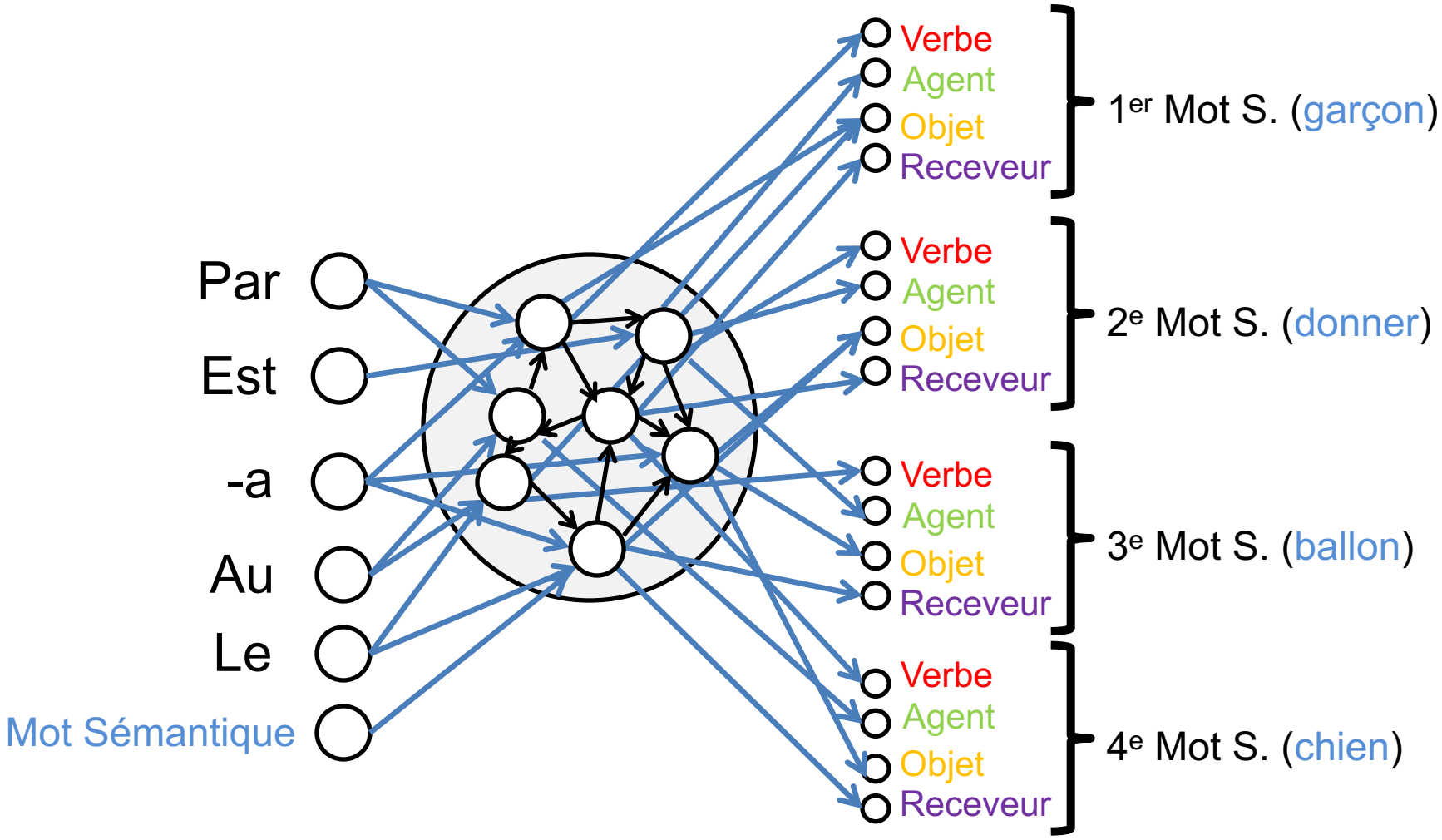



Par
Est
-a
Au
Le
...
Verbe
Agent
Objet
Receveur
1er Mot
Verbe
Agent
Objet
Receveur
2e Mot
Verbe
Agent
Objet
Receveur
3e Mot
Verbe
Agent
Objet
Receveur
4e Mot

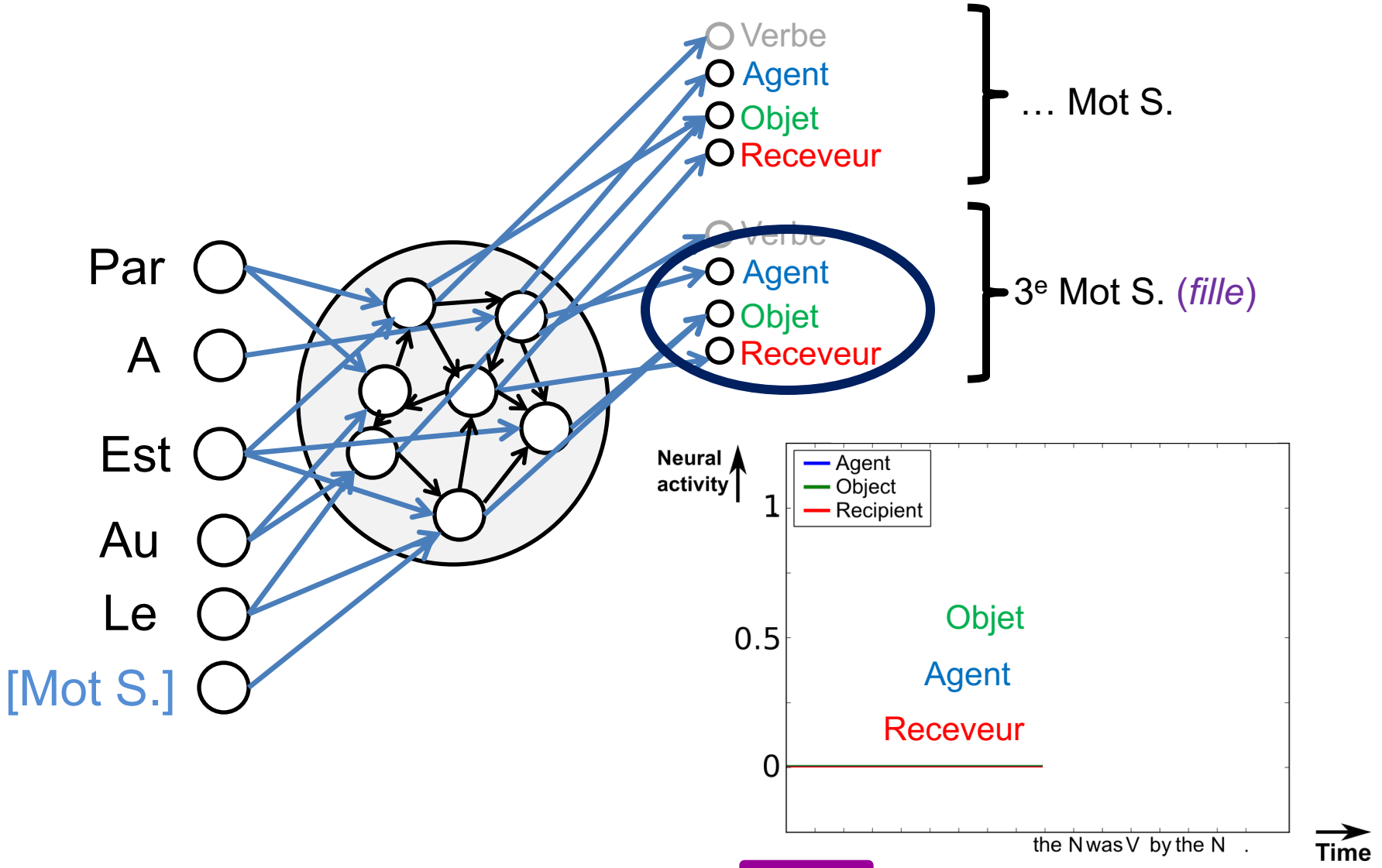

Par
Est
-a
Au
Le
Mot Sémantique
Verbe
Agent
Objet
Receveur
1er Mot Sémantique
Verbe
Agent
Objet
Receveur
2e Mot Sémantique
Verbe
Agent
Objet
Receveur
3e Mot Sémantique
Verbe
Agent
Objet
Receveur
4e Mot Sémantique

Phase d'apprentissage

« Le garçon donn-a le ballon au chien »



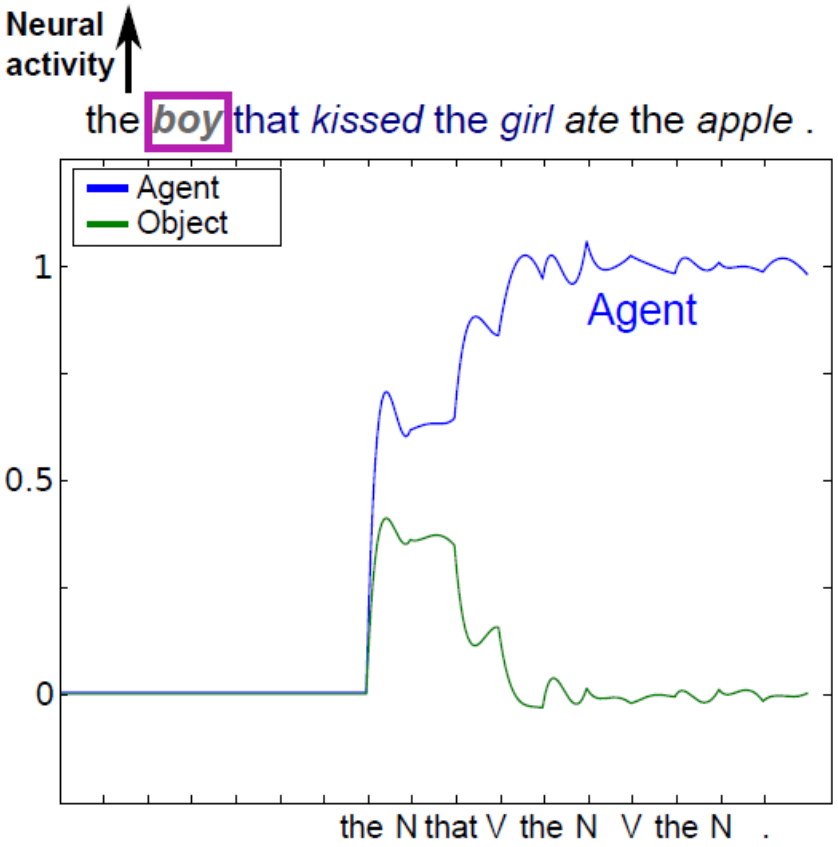
Phase de test



« Le ***chat*** a été ***nourri*** par la ***fille*** »

More complex sentences

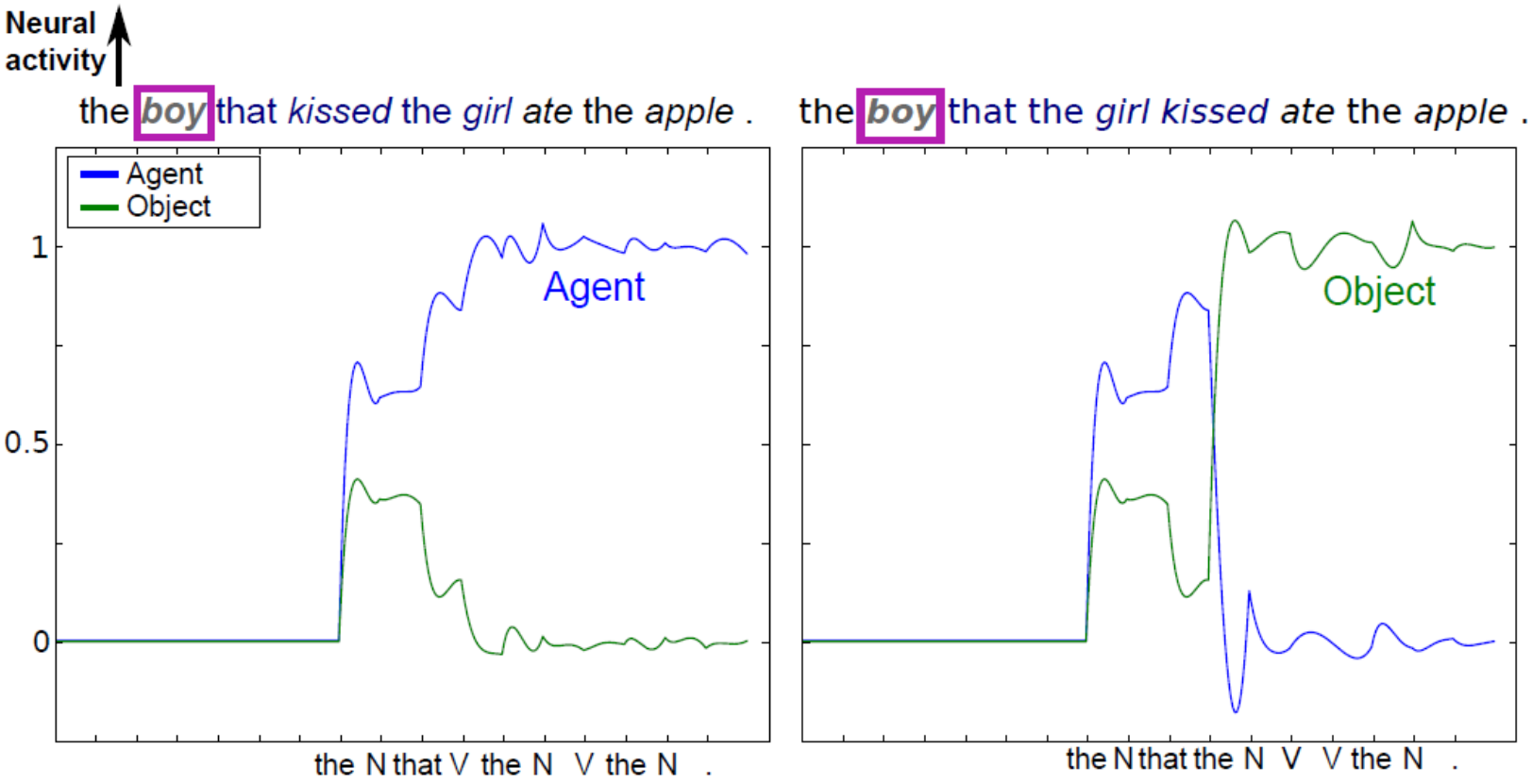
- On-going roles for the relative clause for the 1st SW: **"boy"**
  - The boy that kissed the girl ate the apple. (subject-relative)



Hinaut & Dominey 2013

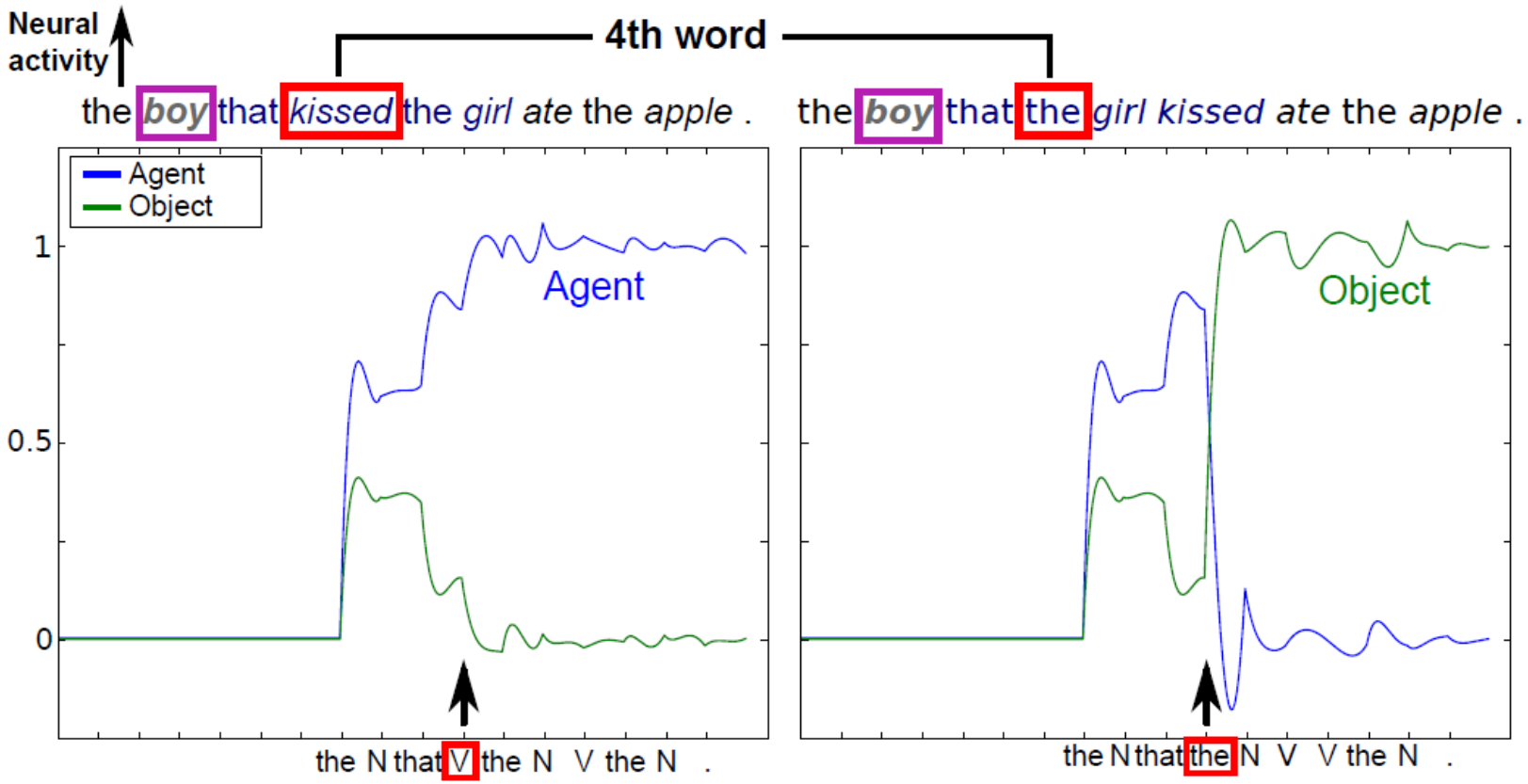
More complex sentences

- On-going roles for the relative clause for the 1st SW: **"boy"**
  - The boy that kissed the girl ate the apple. (subject-relative)
  - The boy that the girl kissed ate the apple. (object-relative)

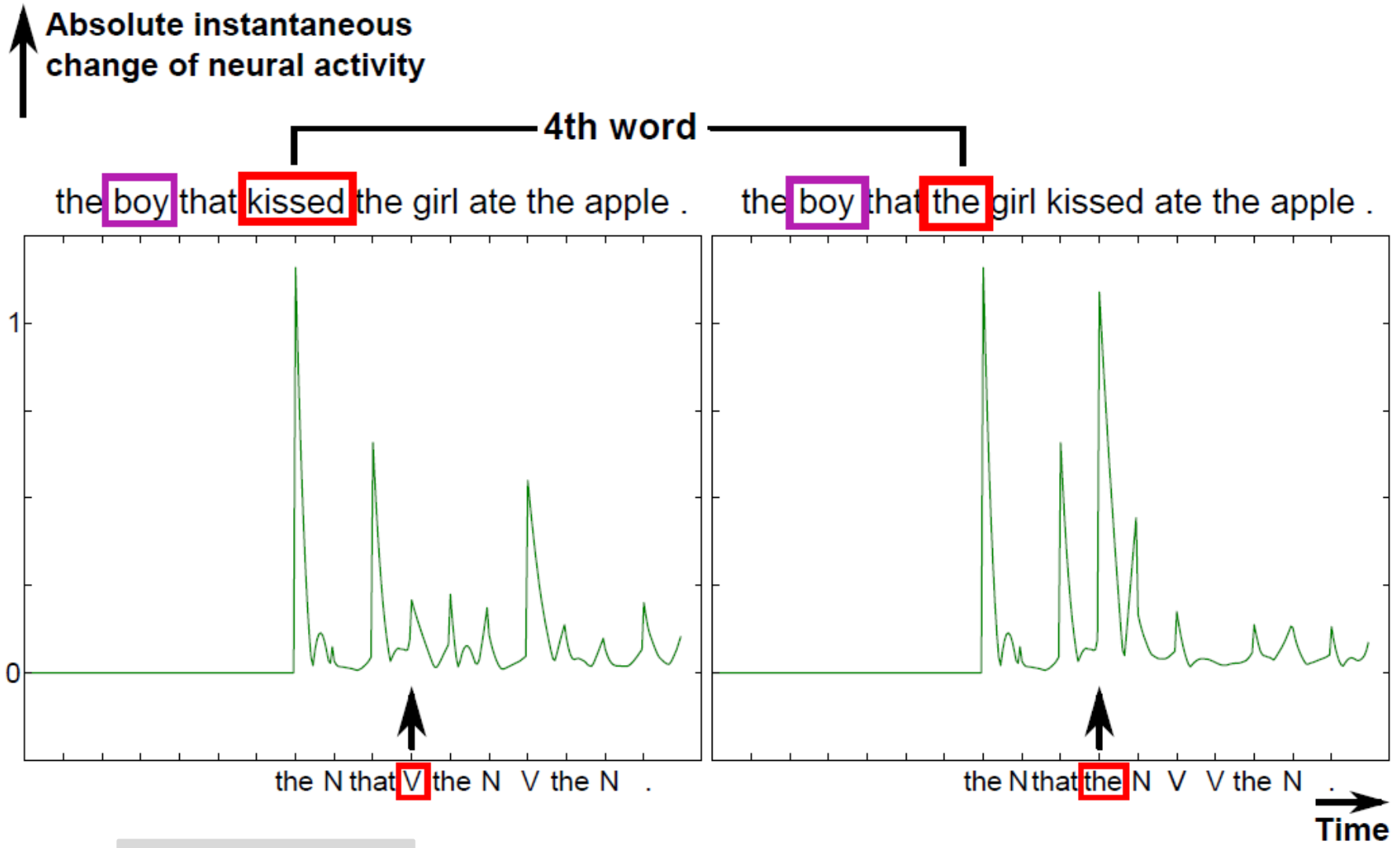


More complex sentences

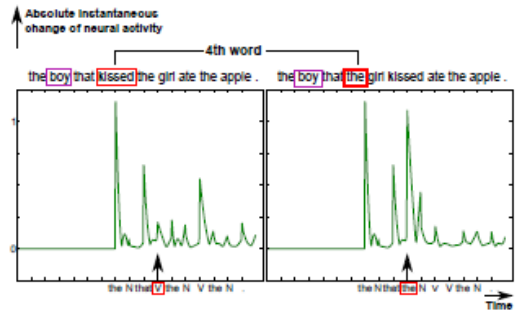
- On-going roles for the relative clause for the 1st SW: **"boy"**
  - The boy that **kissed** the girl ate the apple. (subject-relative)
  - The boy that **the** girl kissed ate the apple. (object-relative)

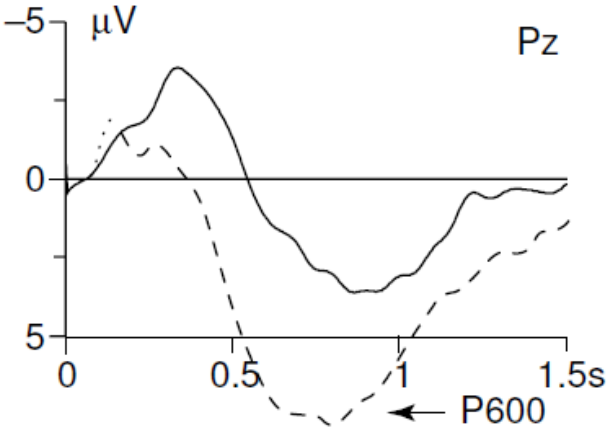


More complex sentences

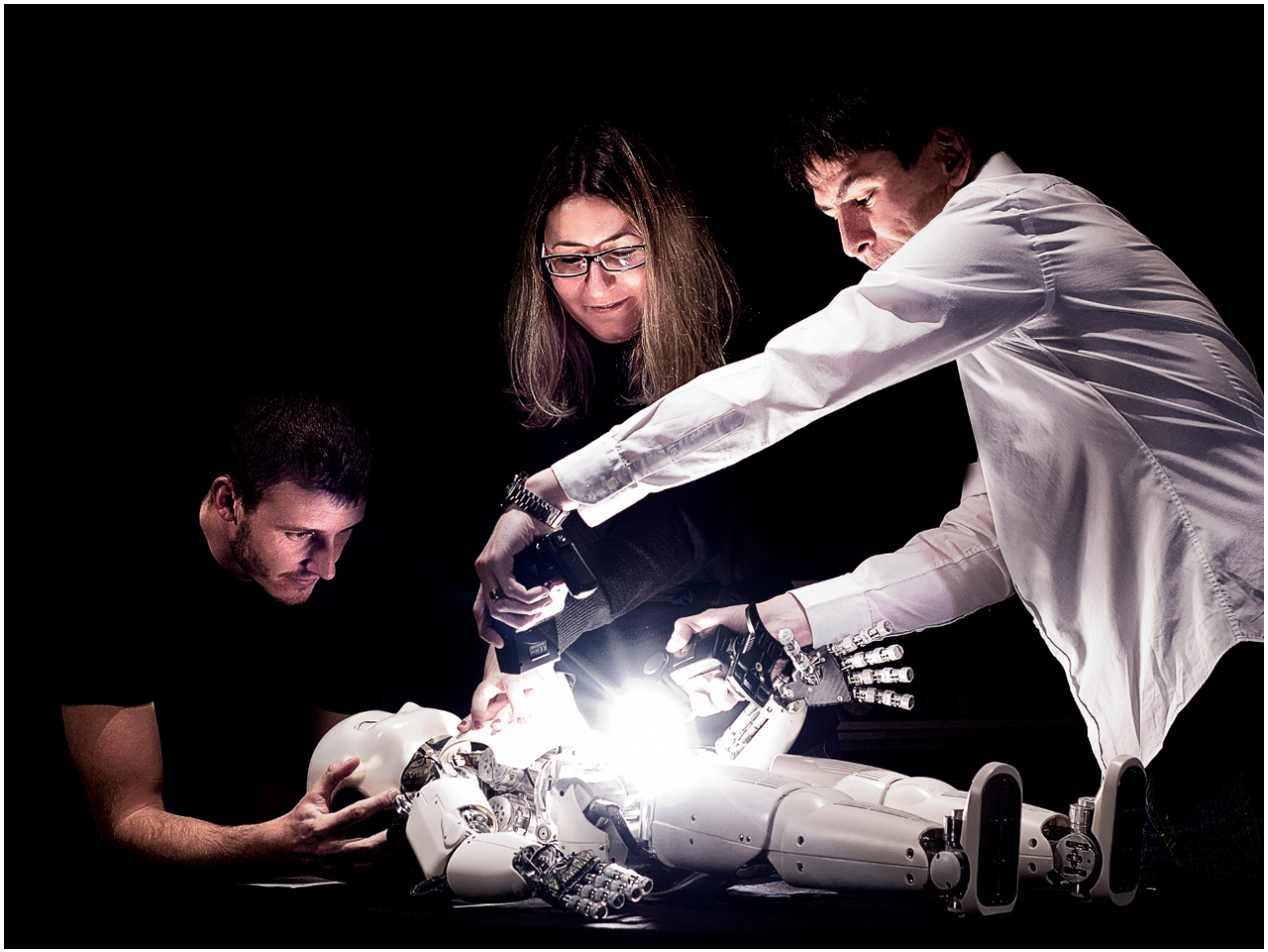


More complex sentences

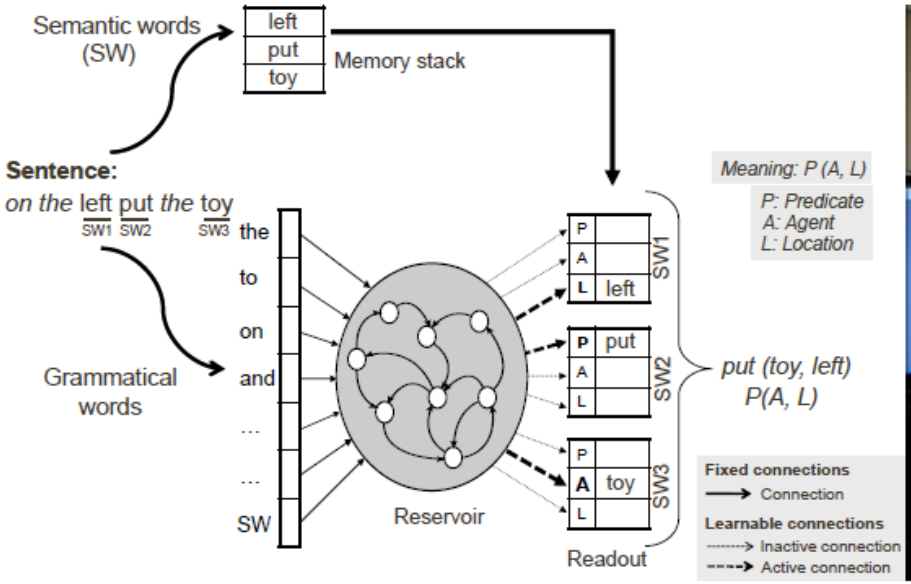




- In humans:
  - Words indicating low frequency constructions can evoke a P600
- In the model:
  - Similarly, a significant change in output activity occurs at the onset of the word indicating unfrequent constructions



Sentence Comprehension for HRI



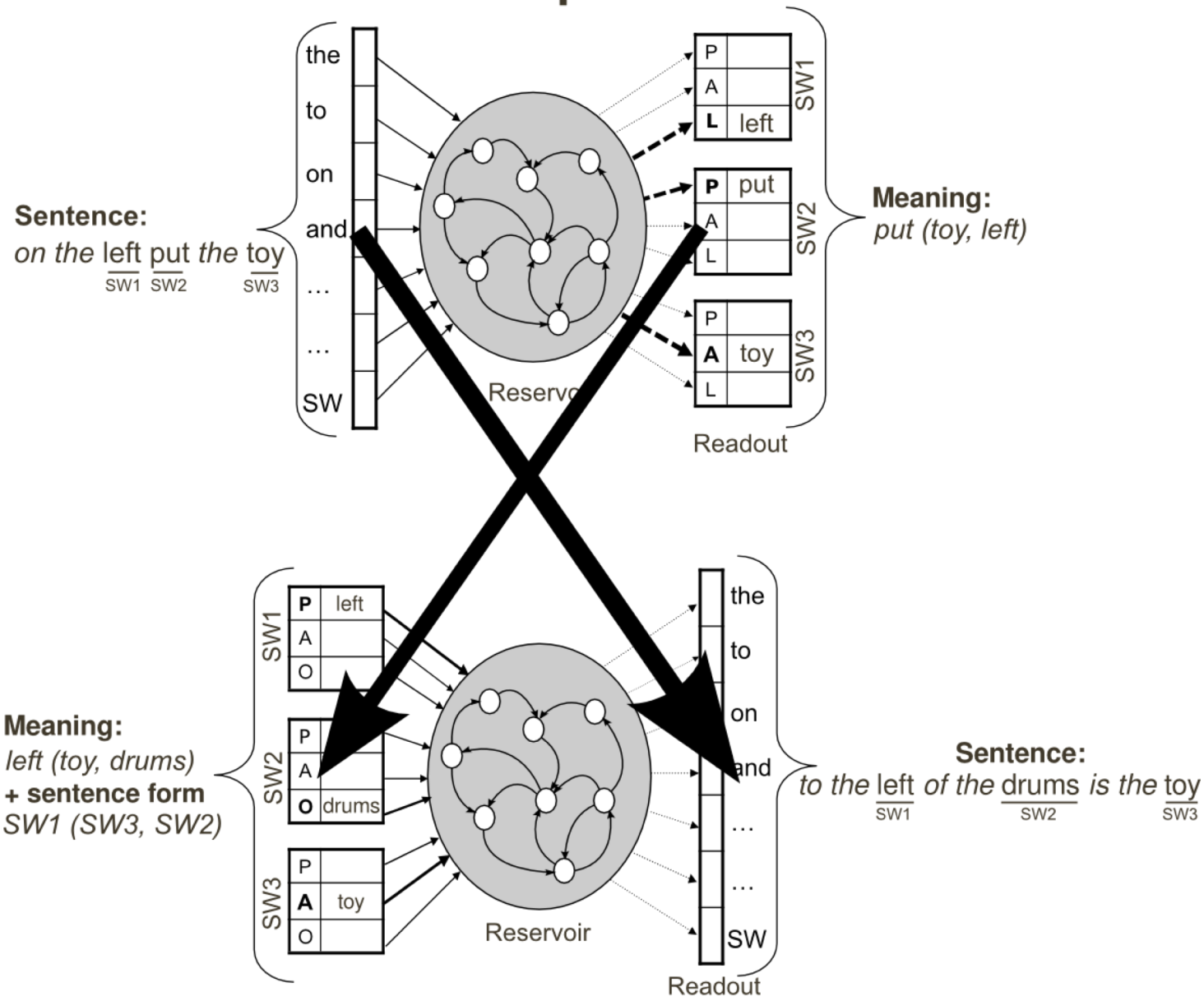
Hinaut et al. 2014

- Two modes: teaching mode and testing mode
- Testing mode:
  - Human speech → Automatic speech recognition (Sphinx-II)
  - → Reservoir processing → Robot (Yarp)

Sentence Comprehension for HRI

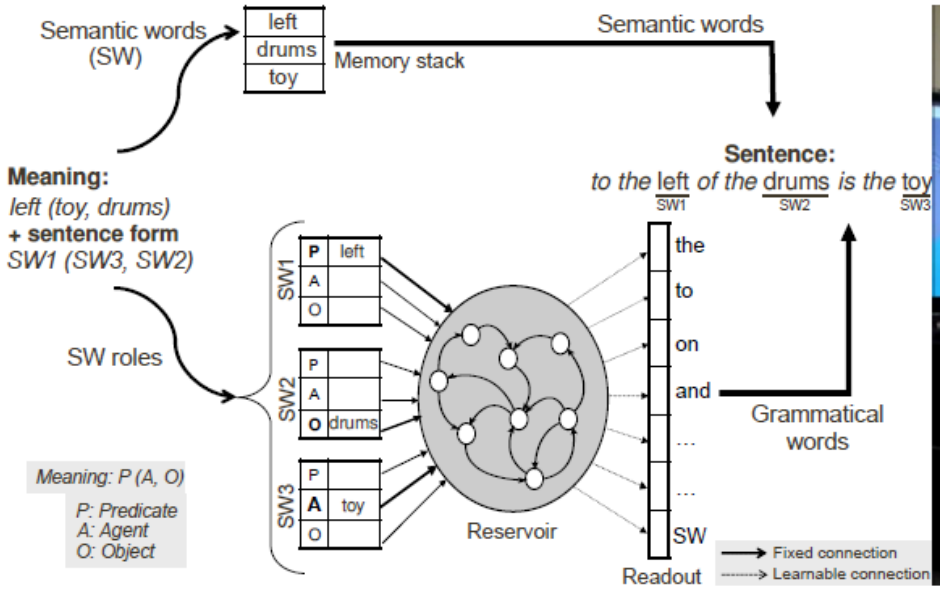
https://youtu.be/AUbJAupkU4M

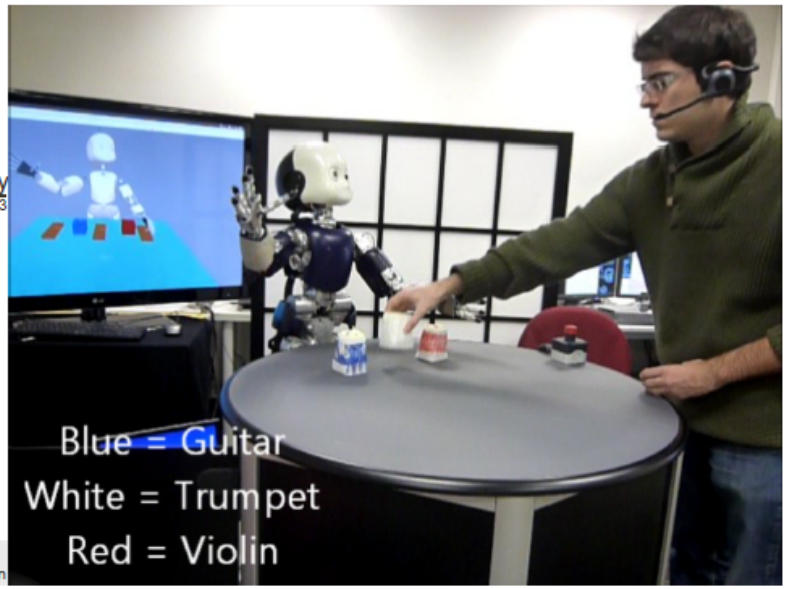
Sentence comprehension model



Sentence production ("inverse") model

Sentence Production



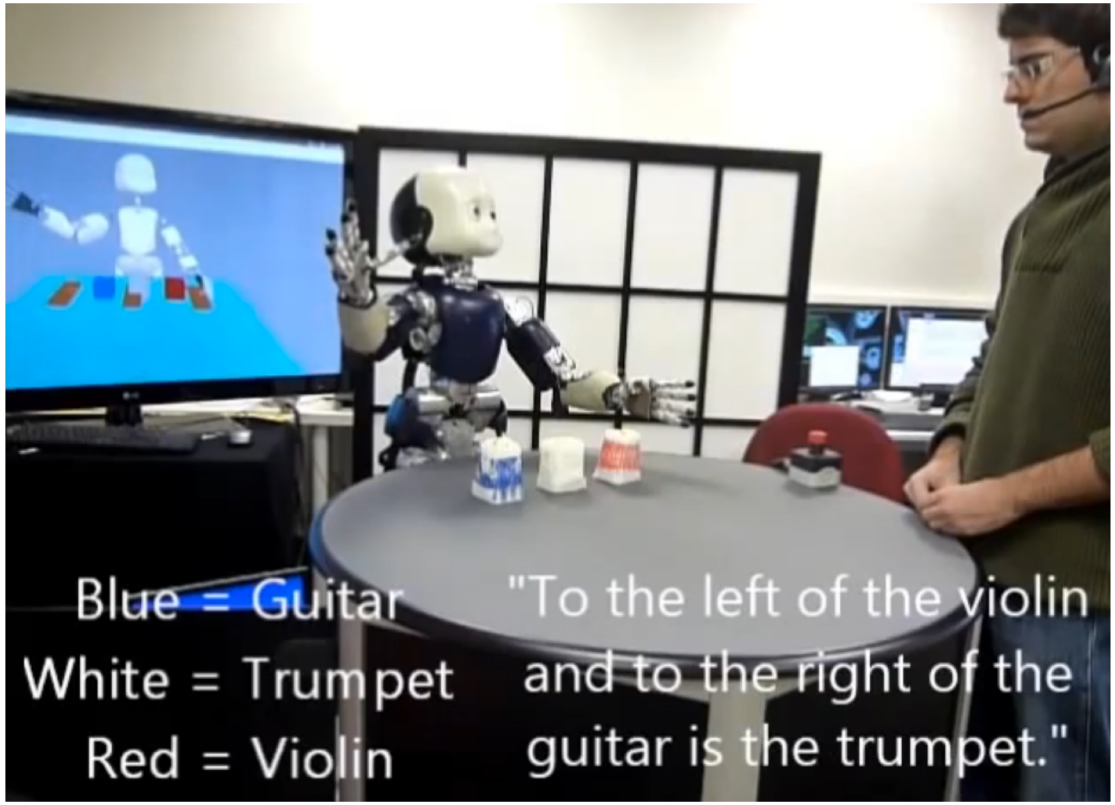


Start movie

- Two modes: teaching mode and testing mode
- Testing mode:
  - User places an object on the table (focus object)
  - → User gives a sentence form (usual/non-usual)
  - → Robot (Yarp) → Reservoir processing → Speech synthetizer

Sentence Production

https://youtu.be/3ZePCuvygi0



Phrases données par des utilisateurs

- point the triangle before grasping the circle
- put the cross to the left before grasping the circle
- point to the cross twice
- before you grasp the cross please grasp the triangle
- before pushing the triangle to the middle please push the cross to the right
- grasp the circle and then point to it
- touch the triangle then move it to the left
- the cross touch it
- point to the circle after having grasped it

Hinaut et al. 2014

Petit essai sur 15 langues !

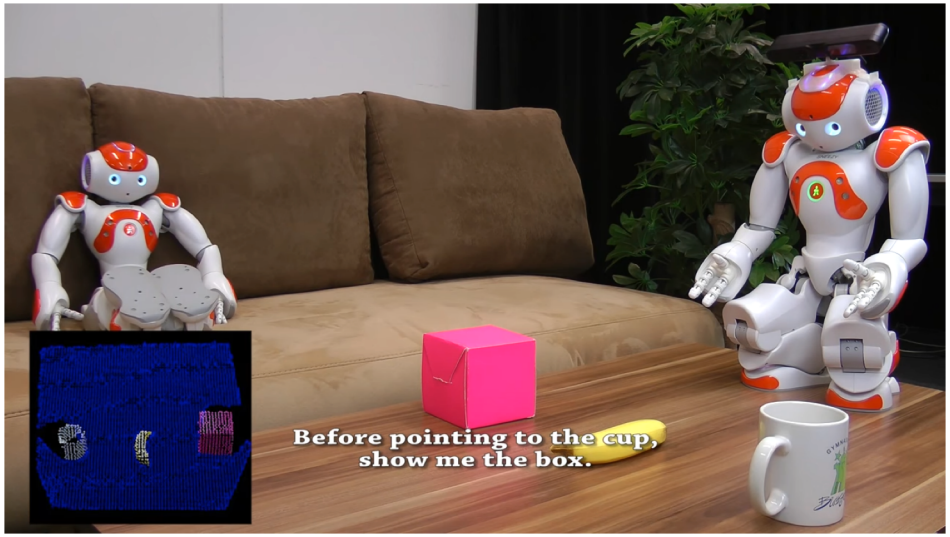
1) answer phone ; answer the phone
2) geh an Telefon ; geh an das Telefon
3) contesta teléfono ; contesta el teléfono
4) réponds téléphone ; réponds au téléphone
5) rispondi telefono ; rispondi al telefono
6) contesta telèfon ; contesta el telèfon
7) 接电话;接那-通电话
8) 接電話;接那-通電話
9) jawab telefon ; jawab -kan panggilan telefon itu
10) cevap ver telefon ; telefon -a cevap ver
11) uchal phone ; phone uchal
12) dho javaab phon ; phon ka javaab dho
13) telefono erantzun ; telefono -a erantzun
14) javab telfon ; telfon ra javab bede
15) atenda telefone ; atenda o telefone
16) вдигни телефон ; вдигни телефон -а

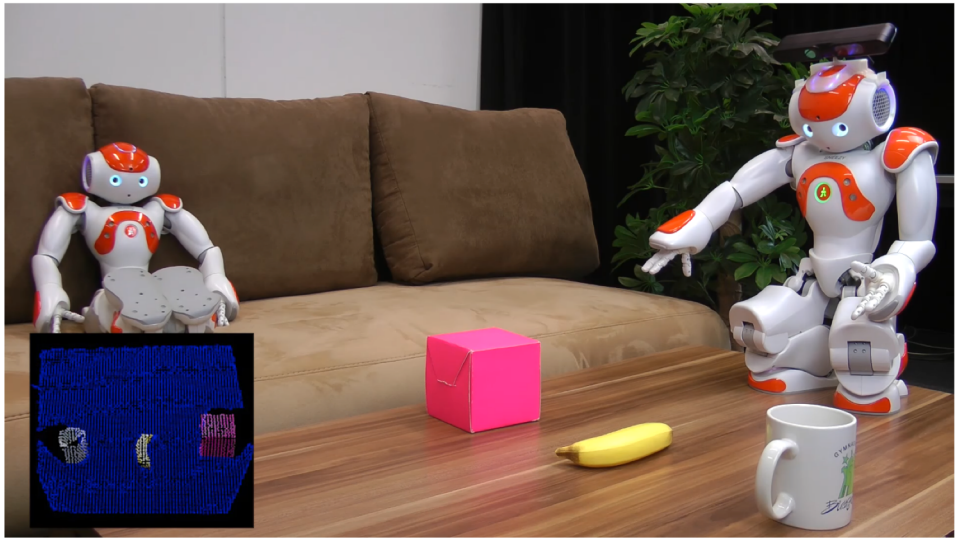
1) English
2) German
3) Spanish
4) French
5) Italian
6) Catalan
7) Simpl. Mandarin
8) Trad. Mandarin
9) Malay
10) Turkish
11) Marathi
12) Hindi
13) Basque
14) Persian
15) Portuguese
16) Bulgarian

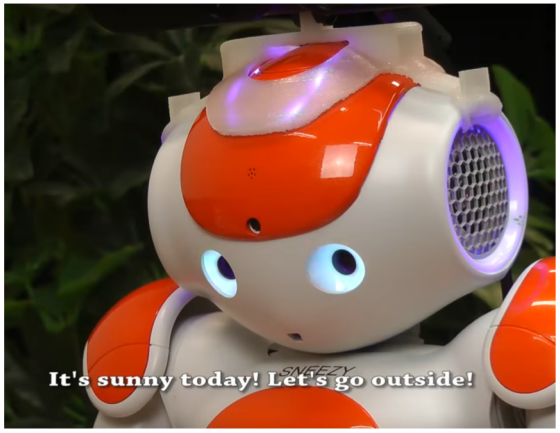
Hinaut & Twiefel 2019

“Nao speaking!”

https://youtu.be/R4cE4bAhLrU







Collaborateurs

- **Lyon**

P.F. Dominey
M. Petit
…

- **Hamburg**

S. Wermter
J. Twiefel
L. Mici
S. Magg
…

- **Paris**

C. del Negro
...

- **Bordeaux**

S. Pagliarini
A. Strock
F. Alexandre
N. Rougier
A. Leblois
…

Campus Paris Saclay
Fondation de Cooperation Scientifique

Organic

Experimental EFAA

MARIE CURIE ACTIONS

RobotDoC
Robotics for Development of Cognition

CAMPUS FRANCE

informatics mathematics Inria

github.com/neuronalX

Tutoriel officiel de spaCy

course.spacy.io

ADVANCED NLP with spaCy

Chapter 1: Finding words, phrases, names and concepts

This chapter will introduce you to the basics of text processing with spaCy. You'll learn about the data structures, how to work with statistical models, and how to use them to predict linguistic features in your text.

Chapter 2: Large-scale data analysis with spaCy

In this chapter, you'll use your new skills to extract specific information from large volumes of text. You''ll learn how to make the most of spaCy's data structures, and how to effectively combine statistical and rule-based approaches for text analysis.

Chapter 3: Processing Pipelines

This chapter will show you to everything you need to know about spaCy's processing pipeline. You'll learn what goes on under the hood when you process a text, how to write your own components and add them to the pipeline, and how to use custom attributes to add your own meta data to the documents, spans and tokens.

Chapter 4: Training a neural network model

In this chapter, you'll learn how to update spaCy's statistical models to customize them for your use case – for example, to predict a new entity type in online comments. You'll write your own training loop from scratch, and understand the basics of how training works, along with tips and tricks that can make your custom NLP projects more successful.