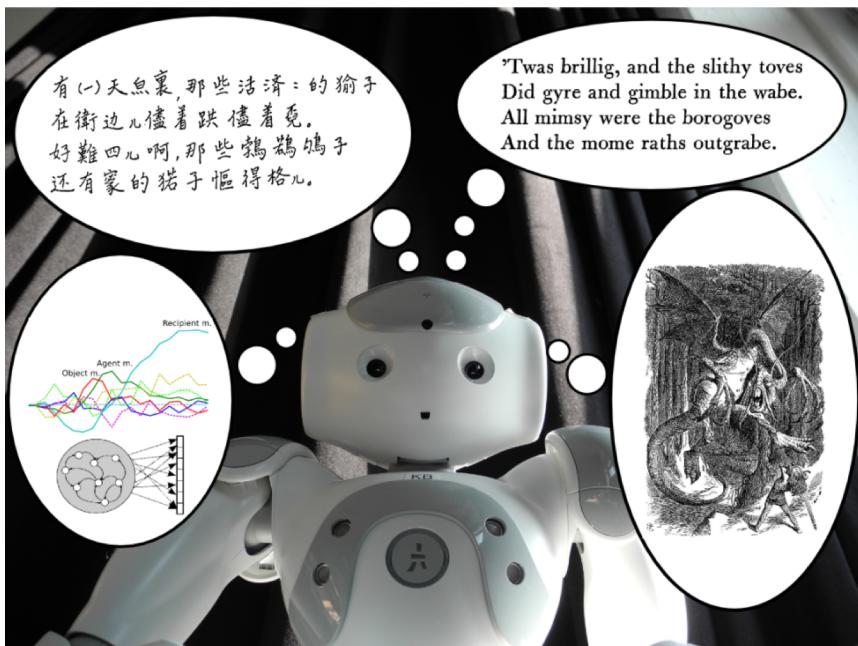


# Une Introduction au Traitement Automatique du Langage

Xavier HINAUT

Chercheur  dans l'équipe

Computational Neuroscience  
Cognition  
**Mnemosyne**  
Brain & Body      Complex systems



@neuronalX  
[github.com/neuronalX](https://github.com/neuronalX)  
[www.xavierhinaut.com](http://www.xavierhinaut.com)  
[xavier.hinaut@inria.fr](mailto:xavier.hinaut@inria.fr)

# Projet github pour le workshop

Partage de liens, données, etc. pour les projets du workshop sur le NLP

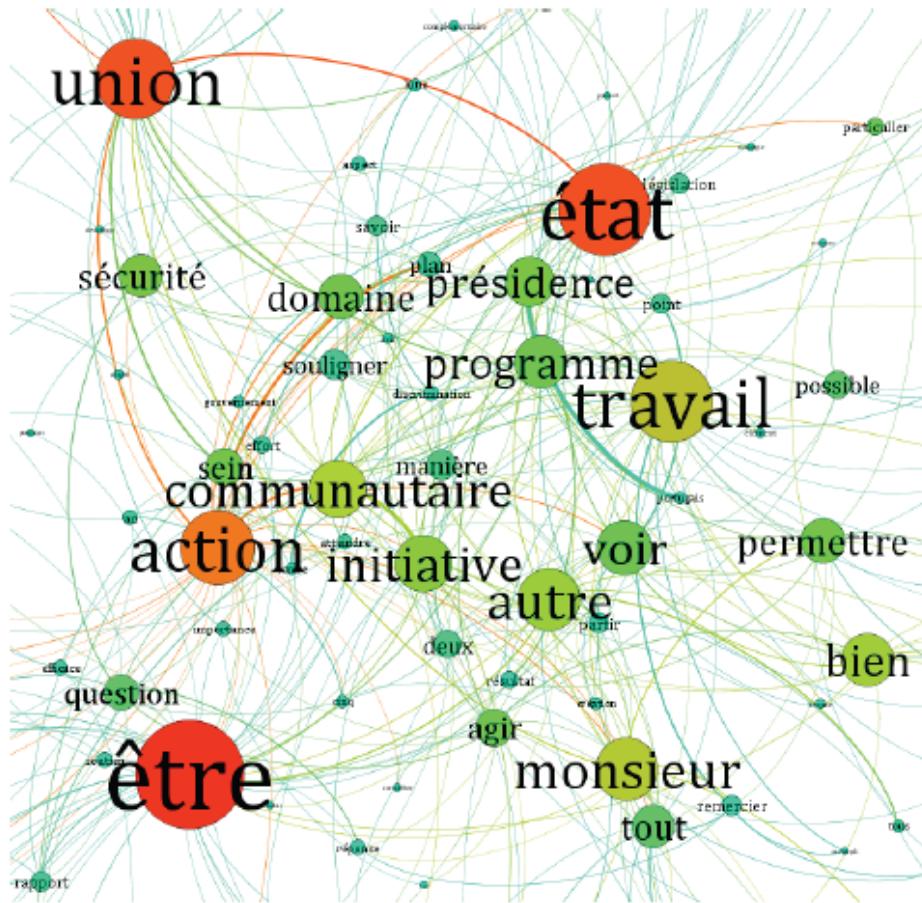
[github.com/neuronalX/workshop-AI4industry-2020](https://github.com/neuronalX/workshop-AI4industry-2020)

« gavagai »



# 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**

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

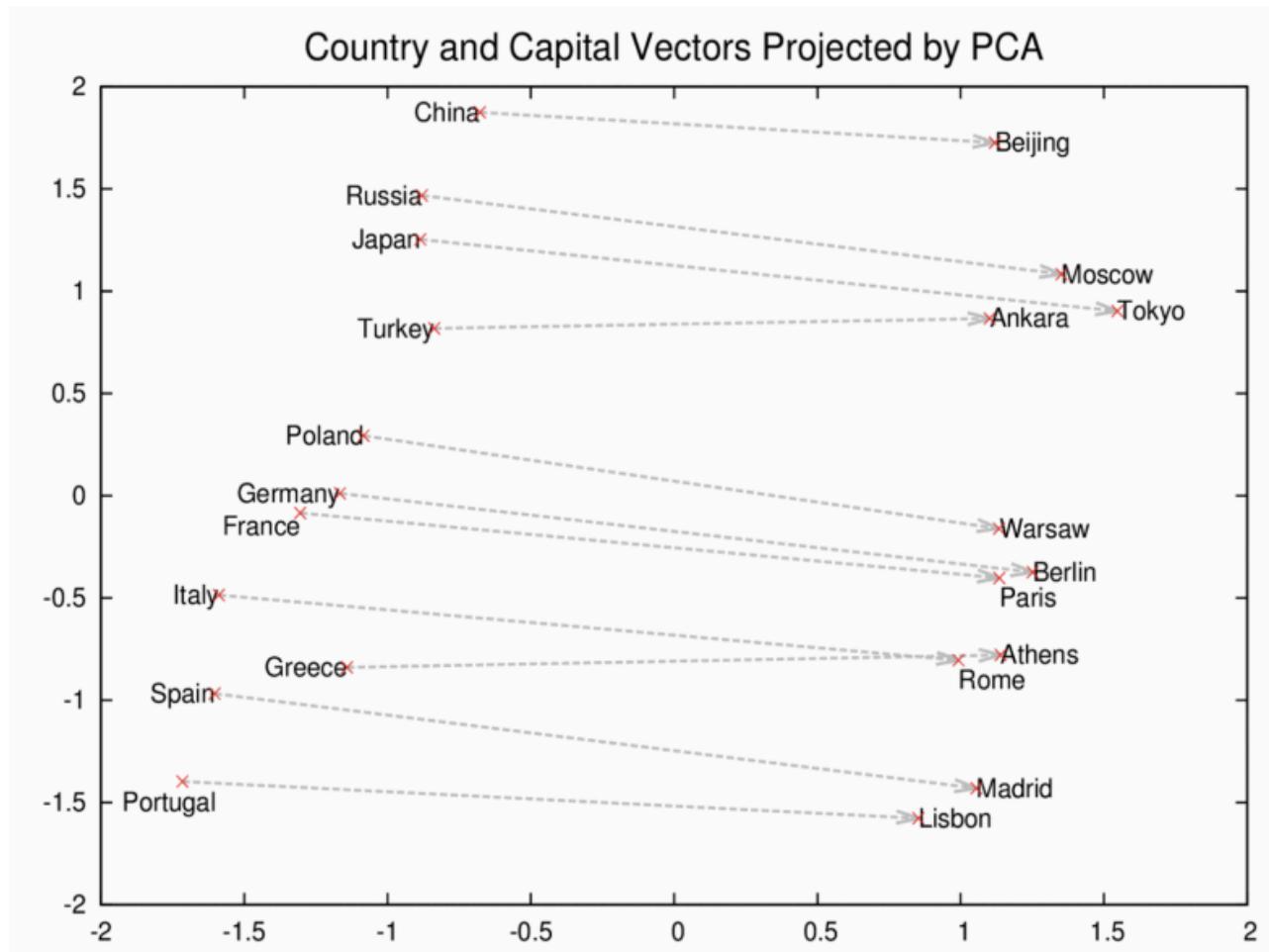
SEARCH

	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 !

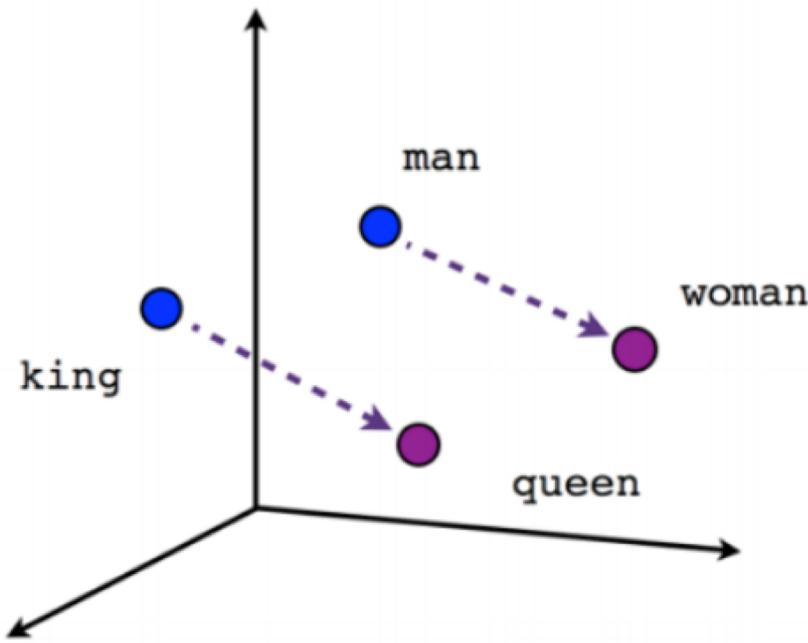
- Sketch Engine:
  - Test word representation:
  - [embeddings.sketchengine.co.uk/static/index.html](http://embeddings.sketchengine.co.uk/static/index.html)
- fastText
  - [github.com/facebookresearch/fastText](https://github.com/facebookresearch/fastText)

# Similarités avec *word2vec*



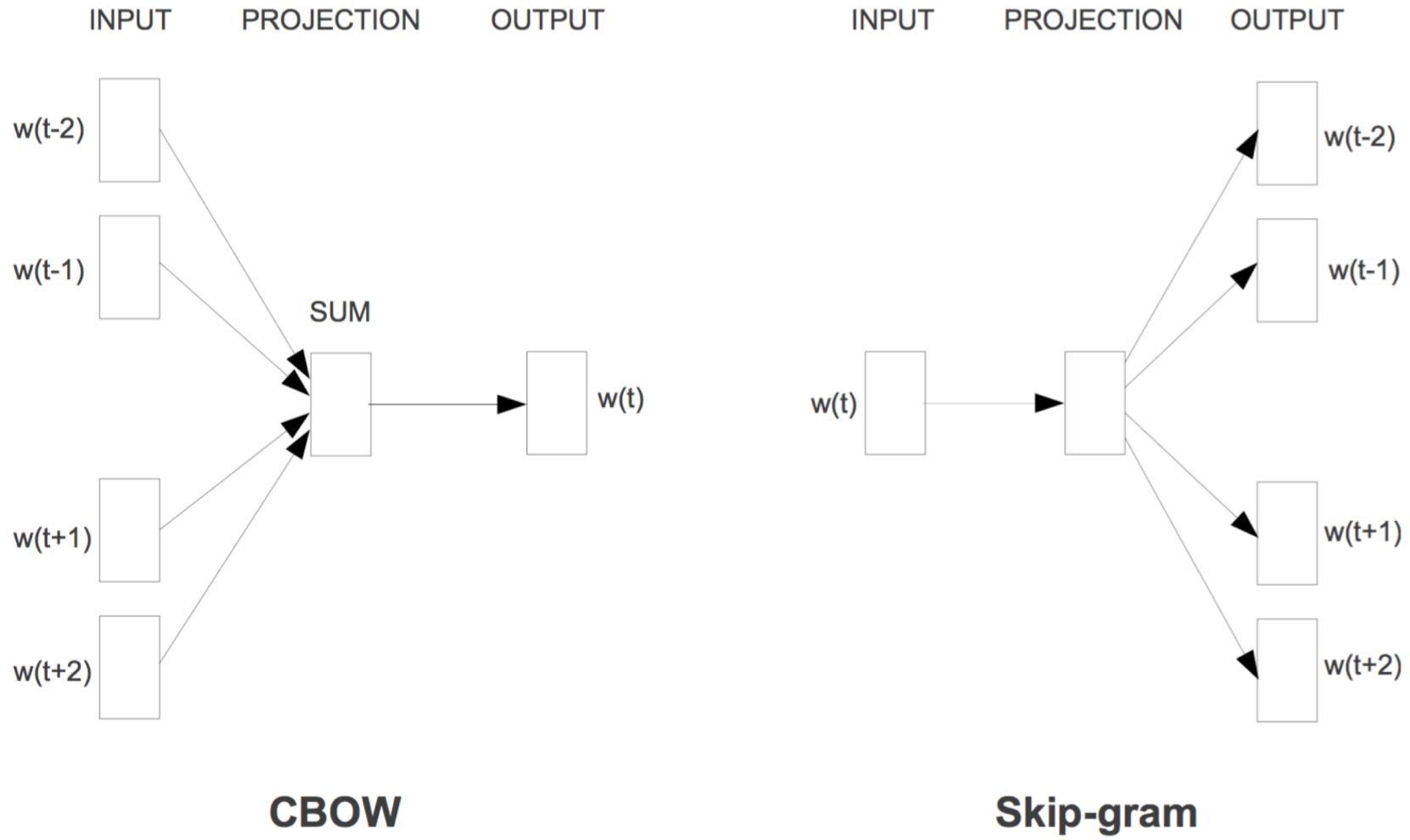
Rome - Italy = Beijing - China

# Combinaisons de vecteurs de mots



roi - homme + femme = reine

# word2vec



**CBOW**

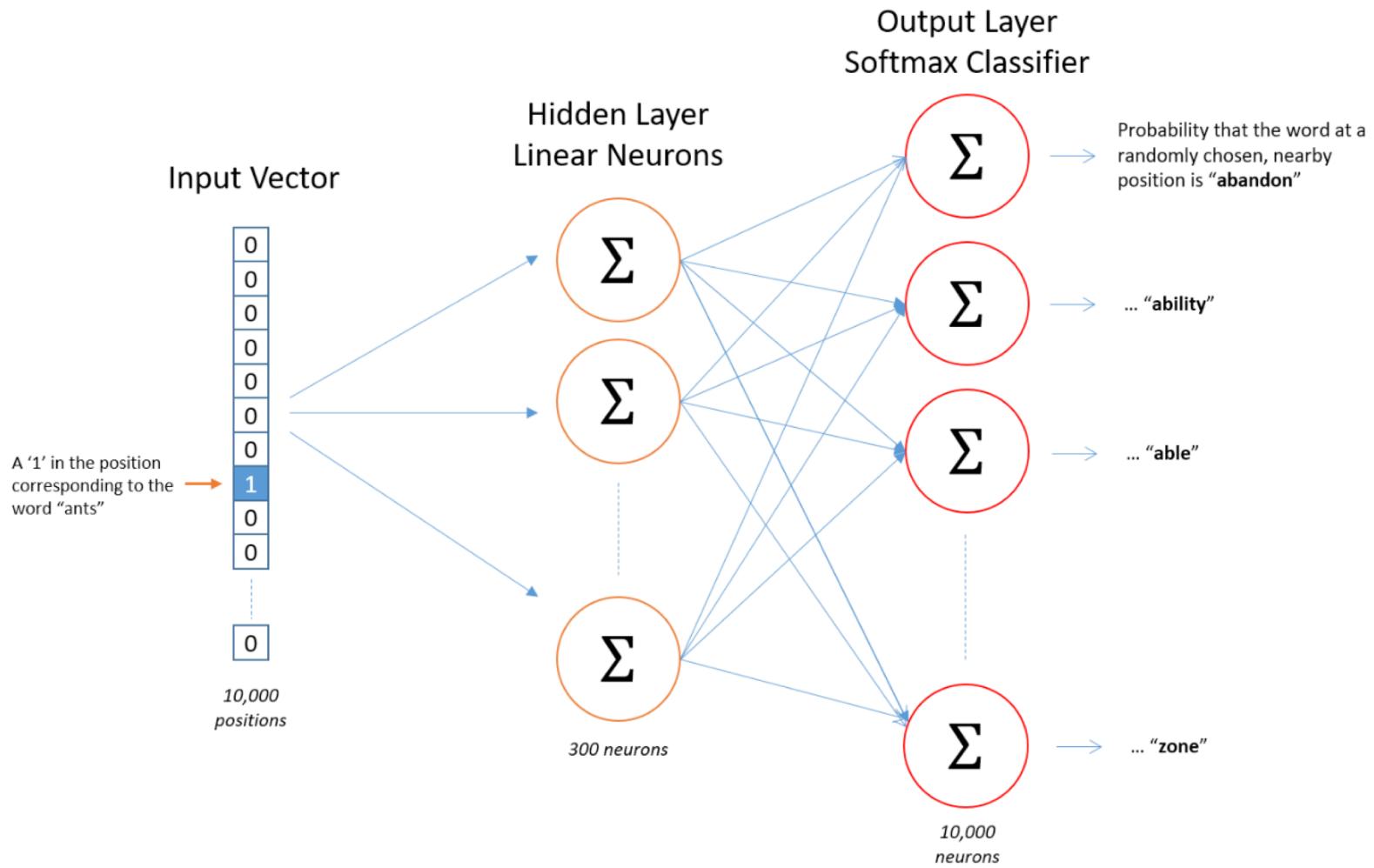
**Skip-gram**

Mikolov et al. 2013

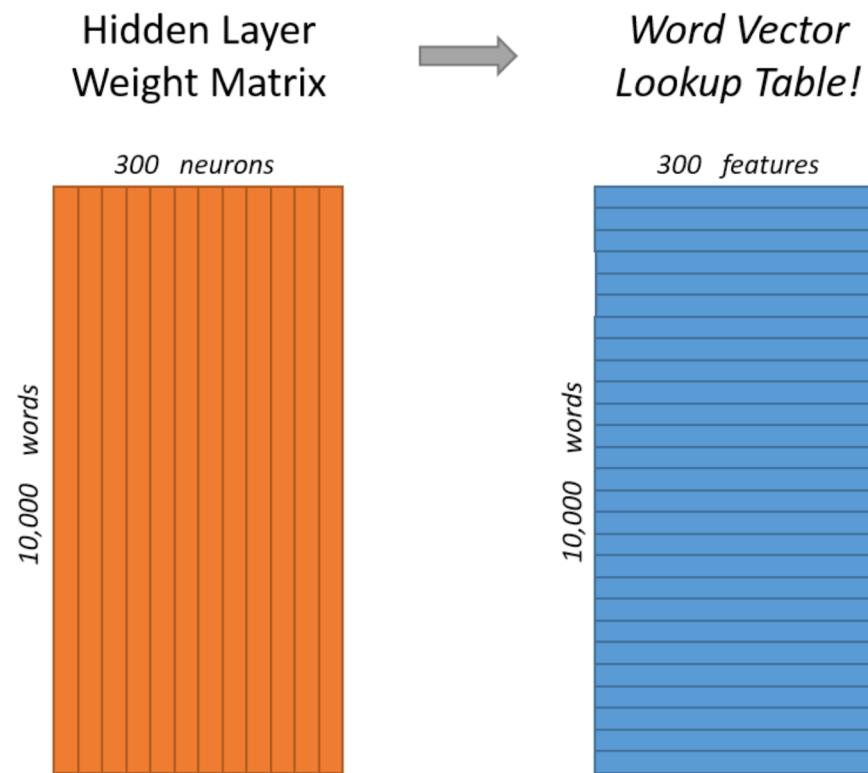
# Word2vec - skip-gram

Source Text	Training Samples
The quick brown fox jumps over the lazy dog. →	(the, quick) (the, brown)
The quick brown fox jumps over the lazy dog. →	(quick, the) (quick, brown) (quick, fox)
The quick brown fox jumps over the lazy dog. →	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
The quick brown fox jumps over the lazy dog. →	(fox, quick) (fox, brown) (fox, jumps) (fox, over)

# Word2vec - skip-gram

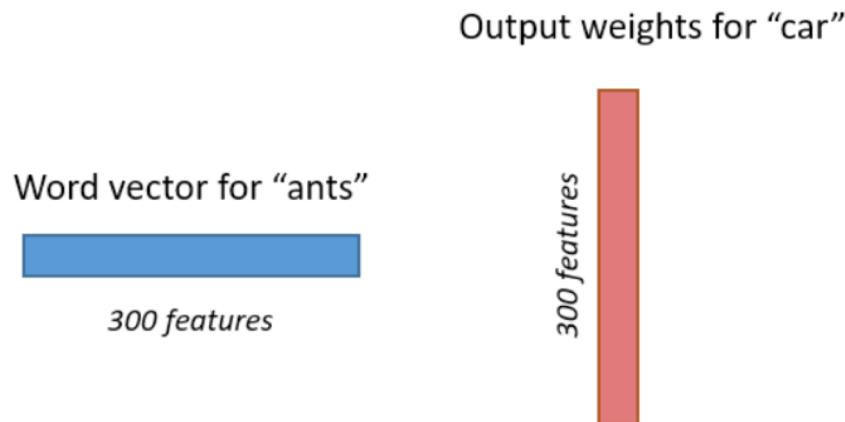


# Word2vec - skip-gram

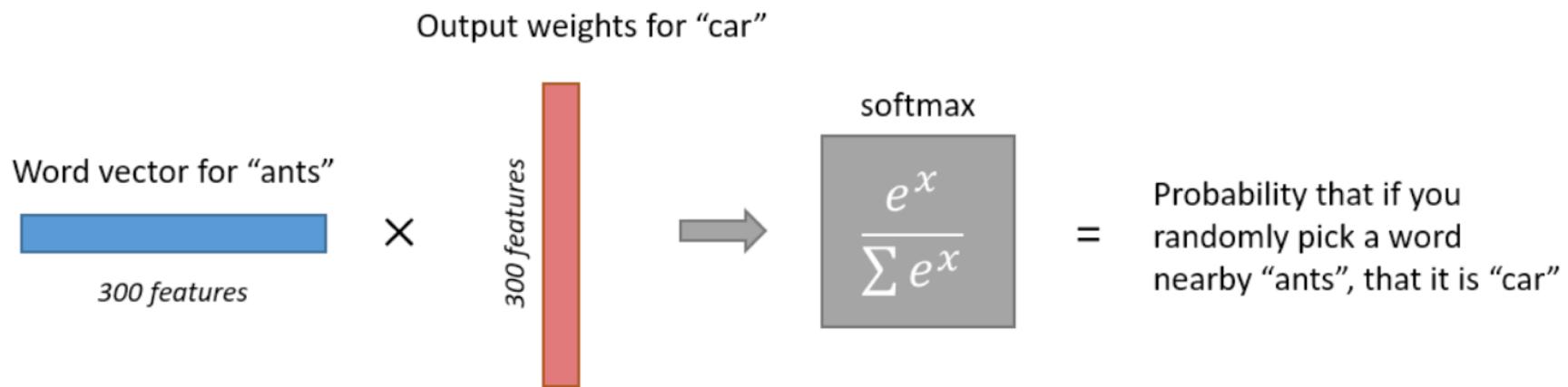


# Calcul de similarité

Comment savoir si deux mots apparaissent dans le même contexte ?



# Calcul de similarité



# Calcul de similarité avec spaCy

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

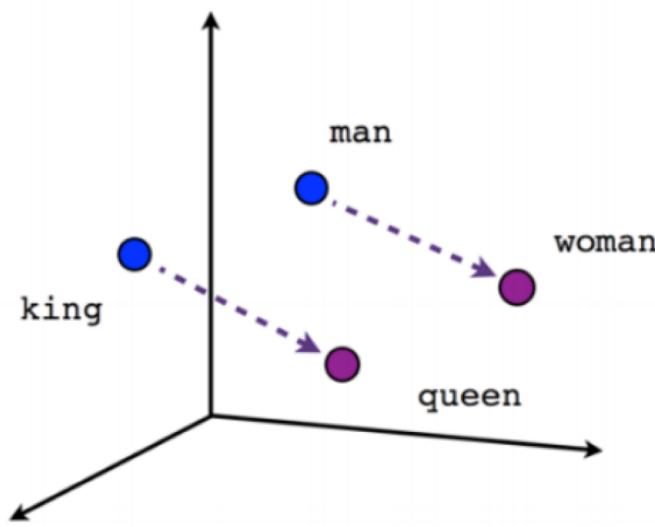
# Compare two documents
doc1 = nlp("I like fast food")
doc2 = nlp("I like pizza")
print(doc1.similarity(doc2))
```

0.8627204117787385

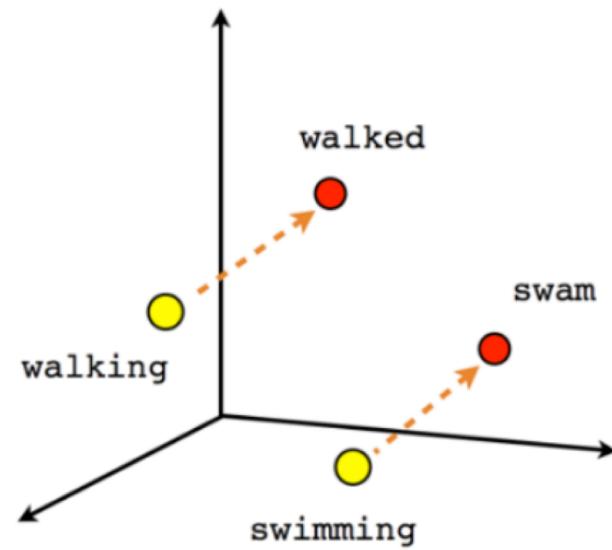
```
# Compare two tokens
doc = nlp("I like pizza and pasta")
token1 = doc[2]
token2 = doc[4]
print(token1.similarity(token2))
```

0.7369546

# Word2vec



Male-Female



Verb tense

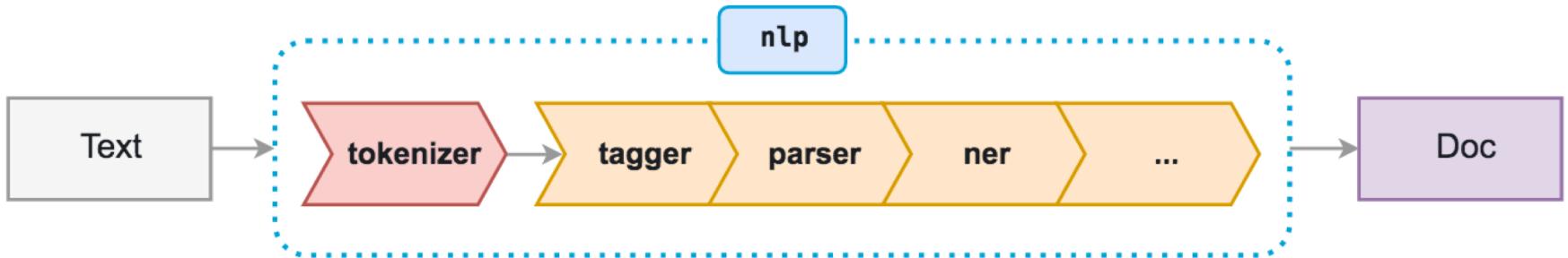
# skip-gram: negative sampling

- 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



**Industrial-Strength  
Natural Language  
Processing**

IN PYTHON

**Get things done**

spaCy is designed to help you do real work — to build real products, or gather real insights. The library respects your time, and tries to avoid wasting it. It's easy to install, and its API is simple and productive. We like to think of spaCy as the Ruby on Rails of Natural Language Processing.

**Blazing fast**

spaCy excels at large-scale information extraction tasks. It's written from the ground up in carefully memory-managed Cython. Independent research in 2015 found spaCy to be the fastest in the world. If your application needs to process entire web dumps, spaCy is the library you want to be using.

**Deep learning**

spaCy is the best way to prepare text for deep learning. It interoperates seamlessly with TensorFlow, PyTorch, scikit-learn, Gensim and the rest of Python's awesome AI ecosystem. With spaCy, you can easily construct linguistically sophisticated statistical models for a variety of NLP problems.

# 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, adjetif, 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
```

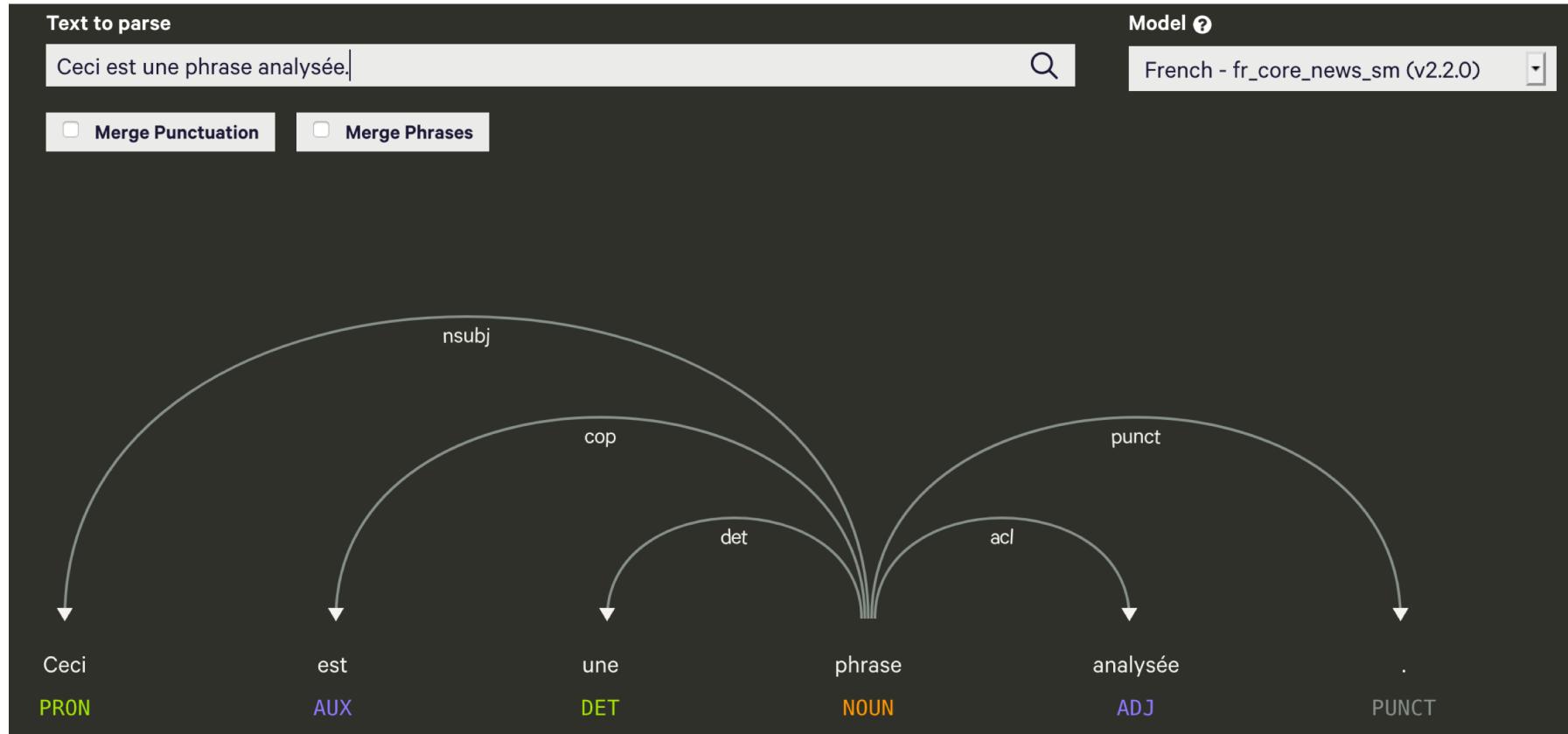
# Named Entity Recognition

- 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.

Apple **ORG** is looking at buying **U.K. GPE** startup for **\$1 billion MONEY**

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

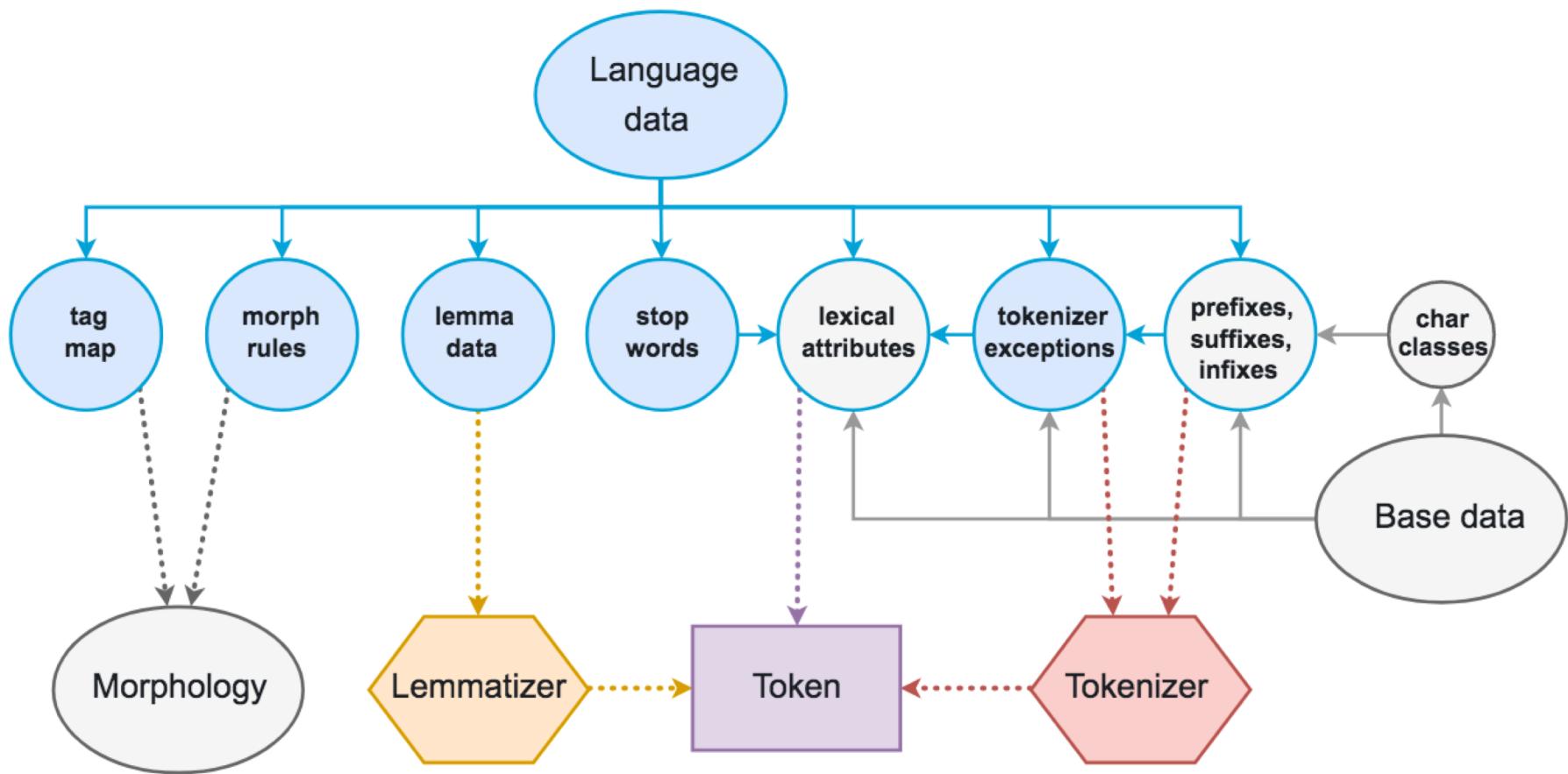


displaCy  
dependency  
visualizer

```
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](http://www.explosion.ai)

# Aperçu des traitements dans spaCy



# Tutoriel officiel de spaCy

[course.spacy.io](https://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.

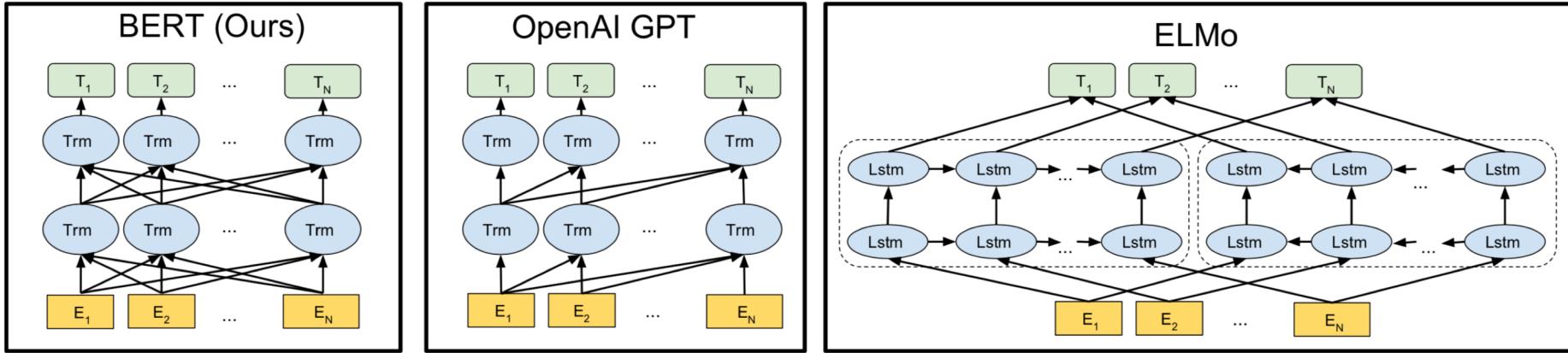
### Chapter 3: Processing Pipelines

This chapter will show you 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



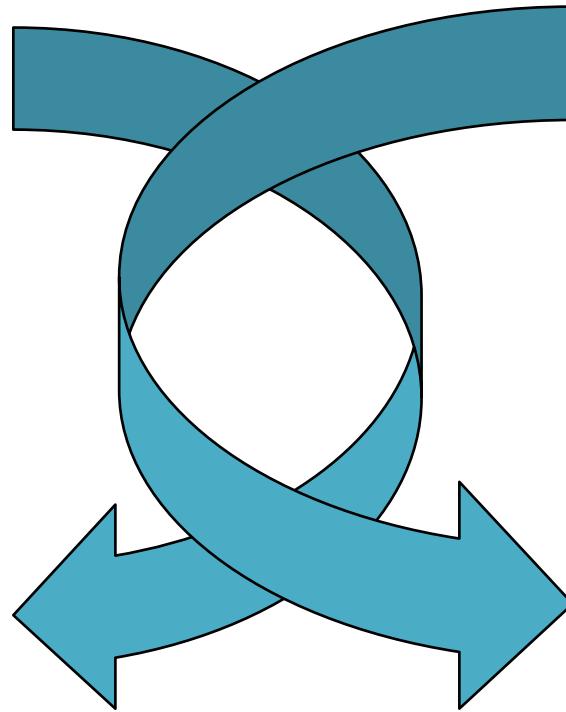
- **BERT** (Delvin et al, 2018, Google AI Language)
  - architecture complexe
  - améliore des problèmes de *word2vec* ou d'autres représentations
    - polysémie
    - ...

# Modéliser l'apprentissage du langage



iCub

**Incarnation  
(ou Ancrage)  
du langage**



**Acquisition  
du langage**

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



iCub

**Ancrage  
du language**



01001100...

“pipe”

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



**Ancrage  
du language**



01001100...



10100110...

“pipe”

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

*Qu'est-ce que la  
“souplesse” ?*

**Ancrage  
du language**

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

*Qu'est-ce que la  
“souplesse” ?*

**Ancrage  
du language**

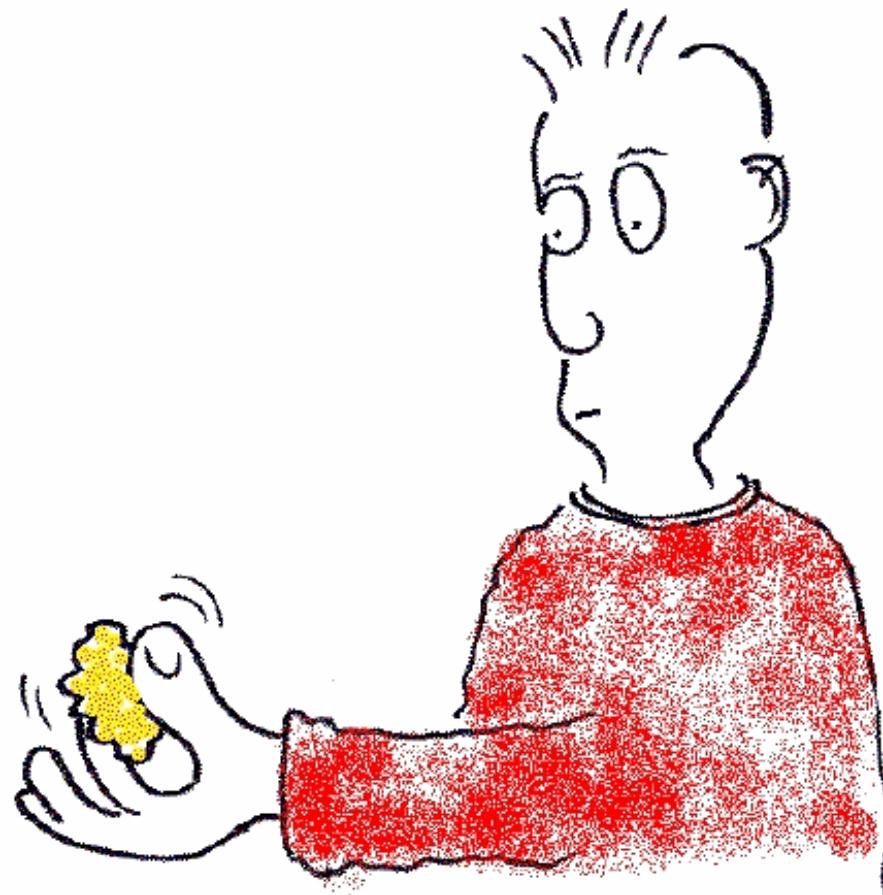
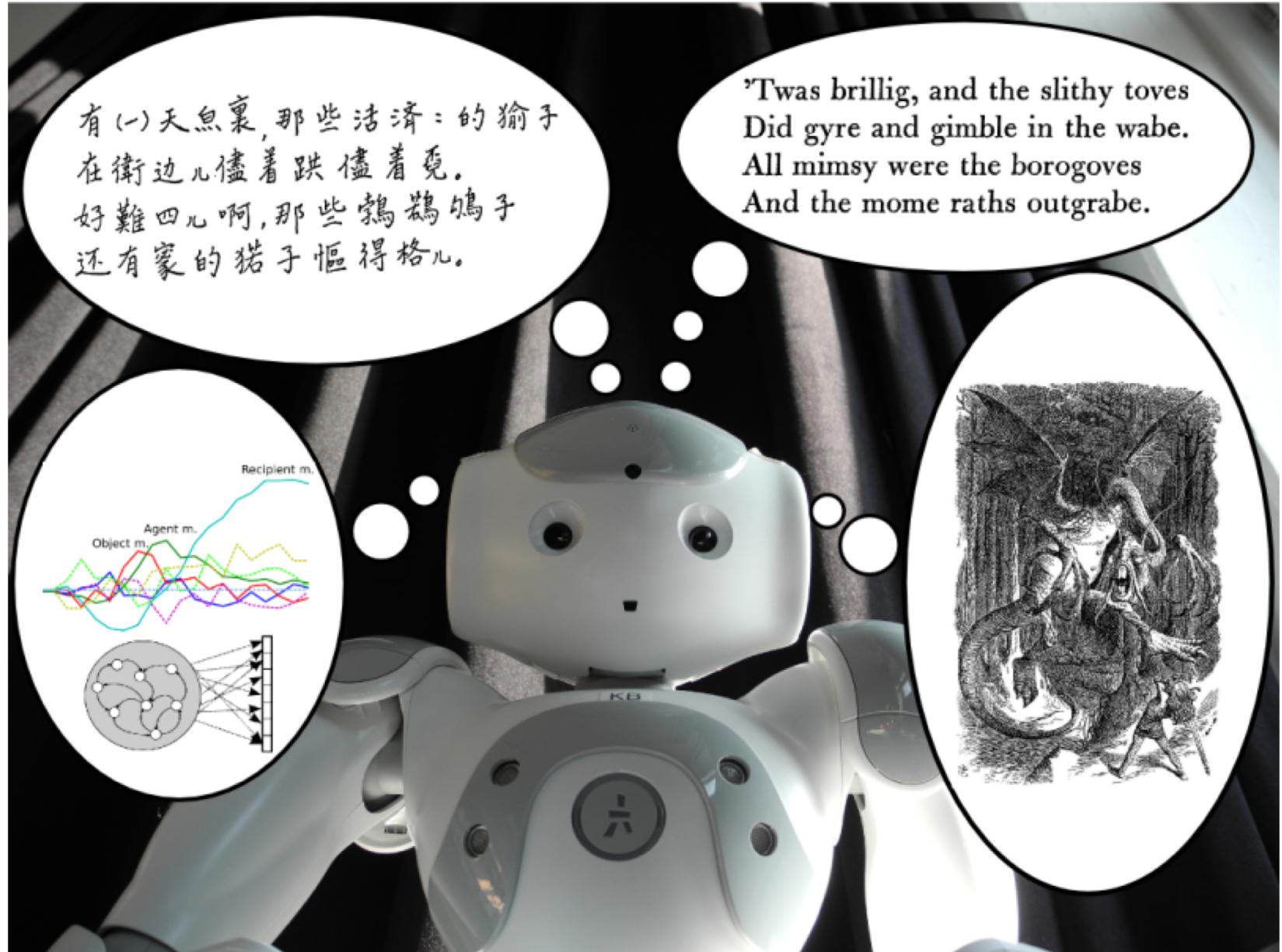
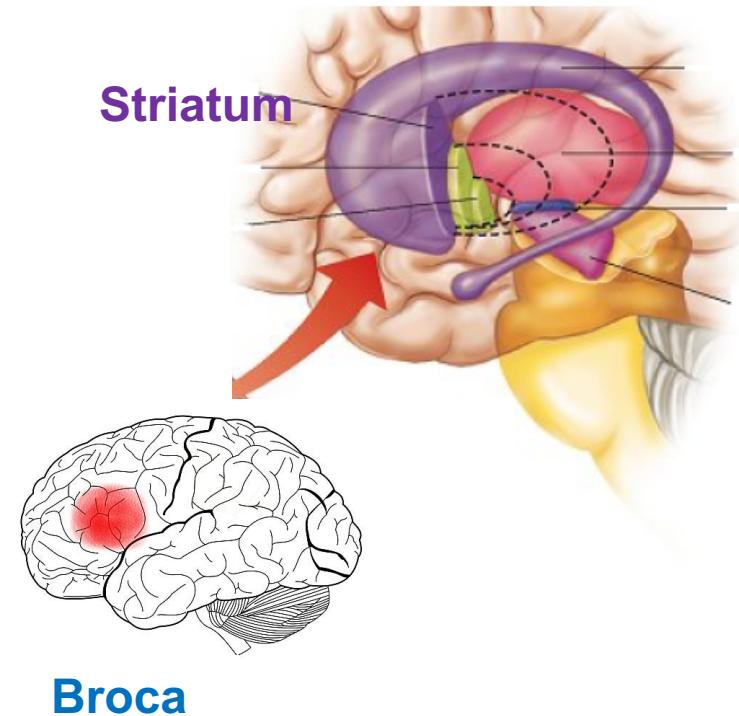
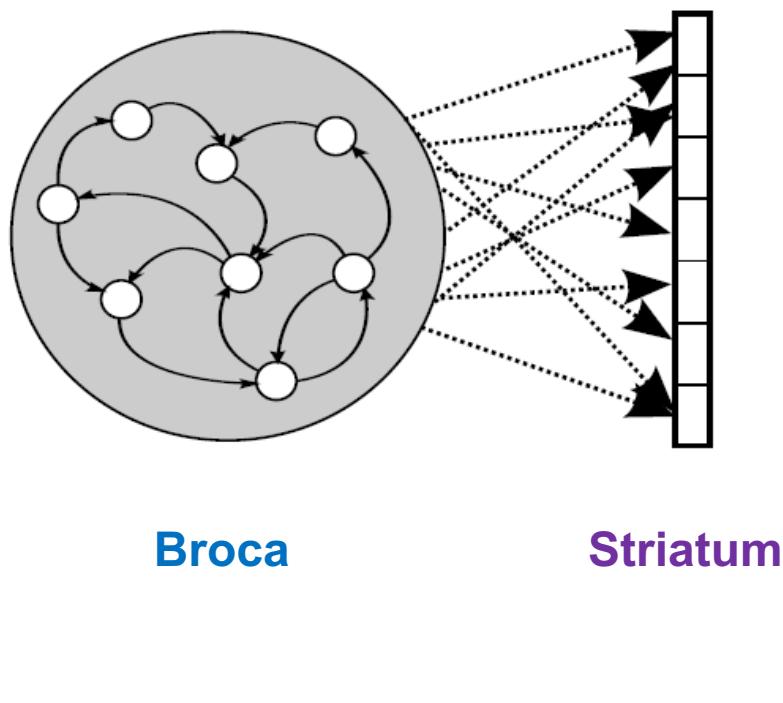


Image: Kevin O'Regan (Why red doesn't sound like a bell?)



# Modéliser l'aire de Broca avec un réservoir



[...] 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 »

```
graph LR; agent[agent] --> garcon[Le garçon]; verbe[verbe] --> donne[donné]; objet[objet] --> ballon[le ballon]; receveur[receveur] --> chien[au chien]
```

# Lier les mots à leur rôle

agent , verbe, objet , receveur

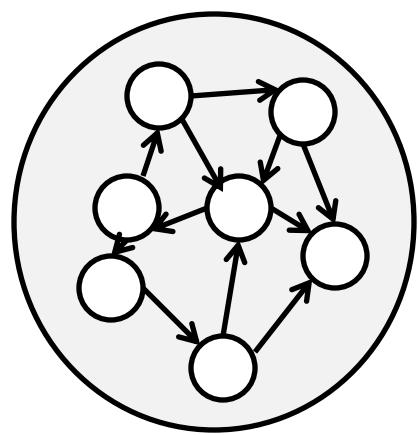
« Le garçon a donné le ballon au chien »

```
graph TD; A[agent] --> garcon[Le garçon]; B[verbe] --> donne[donné]; C[objet] --> ballon[le ballon]; D[receveur] --> chien[au chien]
```

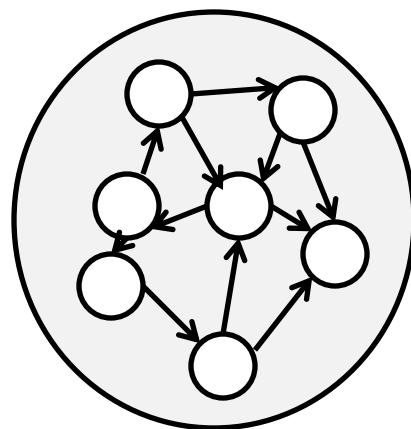
objet , verbe, agent , receveur

« Le ballon est donné par le garçon au chien »

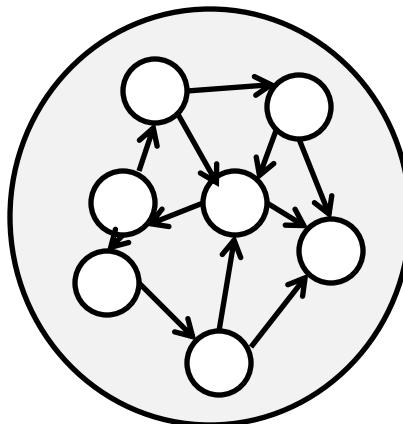
```
graph TD; A[objet] --> ballon[Le ballon]; B[verbe] --> donne[donné]; C[agent] --> garcon[par le garçon]; D[receveur] --> chien[au chien]
```



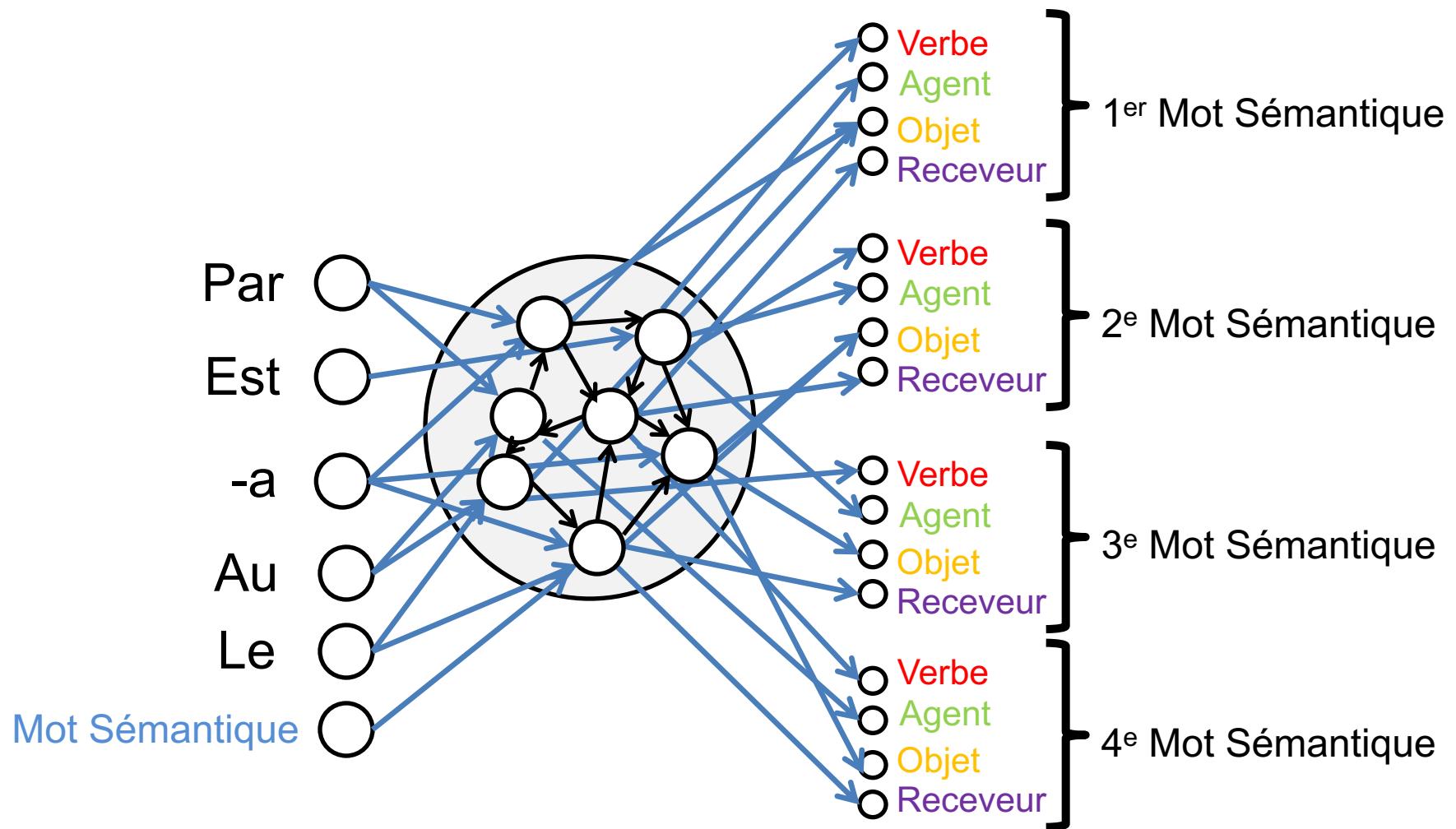
Par ○  
Est ○  
-a ○  
Au ○  
Le ○  
... ○



Par ○  
Est ○  
-a ○  
Au ○  
Le ○  
... ○

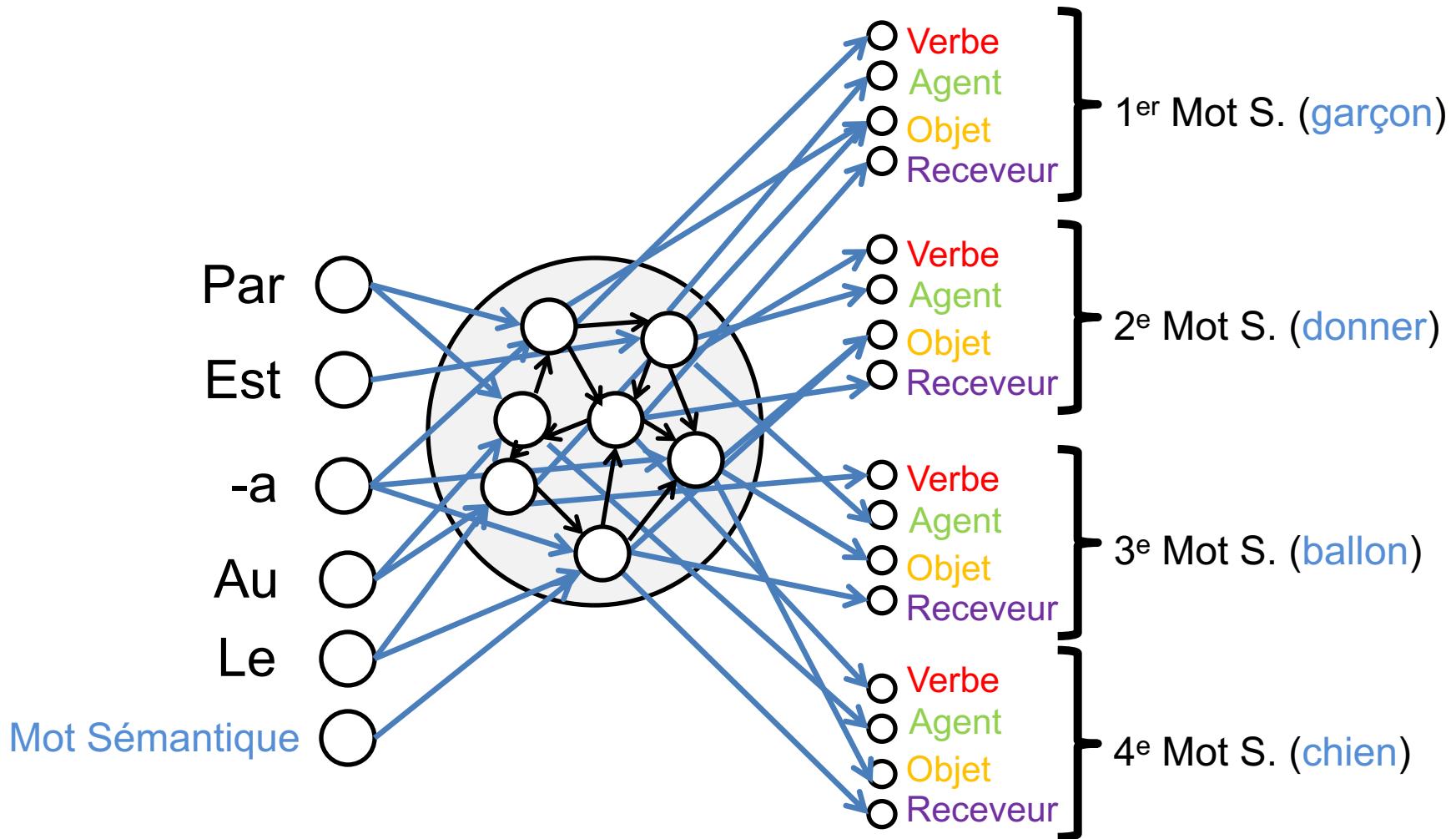


- Verbe
  - Agent
  - Objet
  - Receveur
- ]} 1<sup>er</sup> Mot
- Verbe
  - Agent
  - Objet
  - Receveur
- ]} 2<sup>e</sup> Mot
- Verbe
  - Agent
  - Objet
  - Receveur
- ]} 3<sup>e</sup> Mot
- Verbe
  - Agent
  - Objet
  - Receveur
- ]} 4<sup>e</sup> Mot

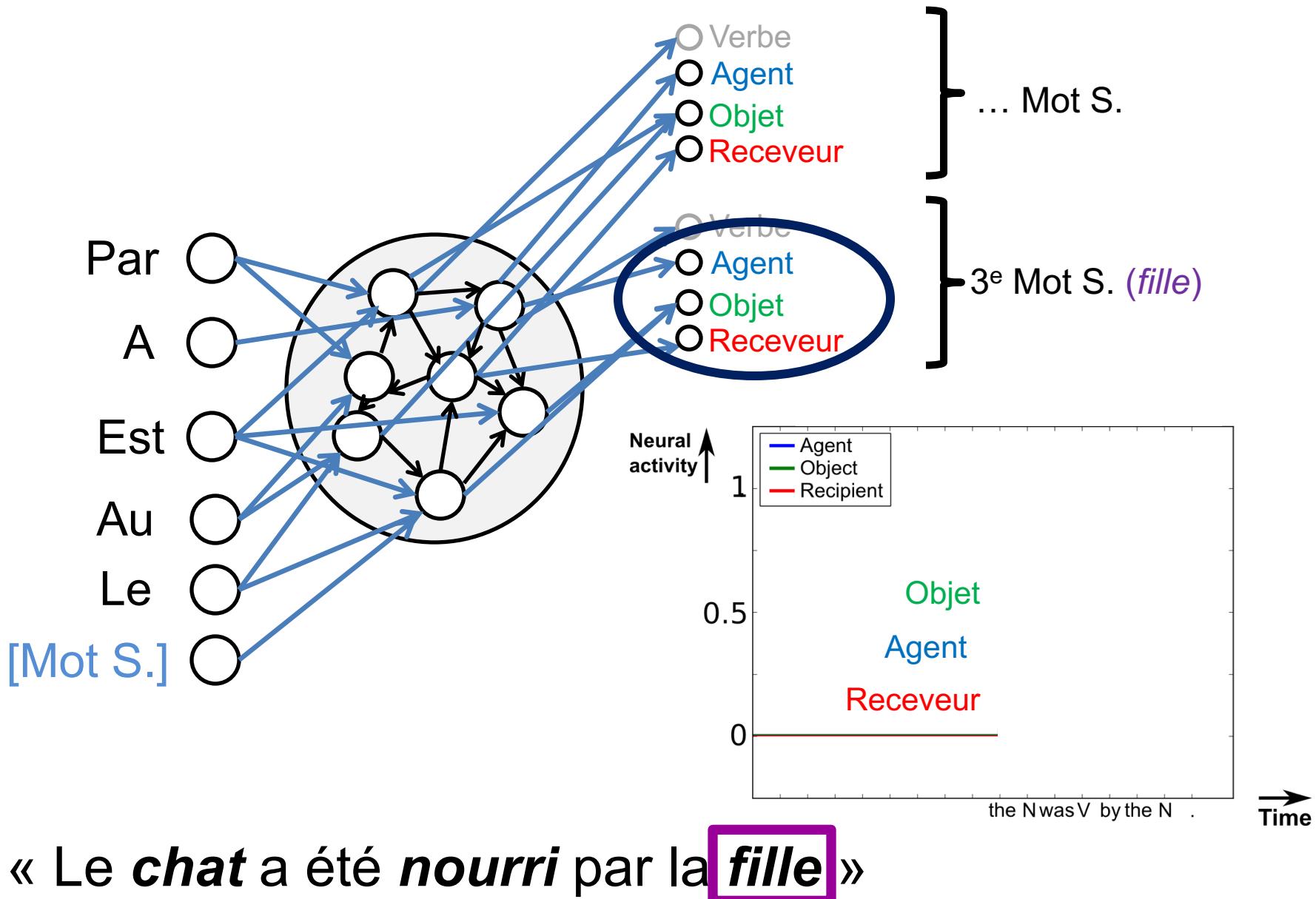


# Phase d'apprentissage

« Le garçon donn-a le ballon au chien »

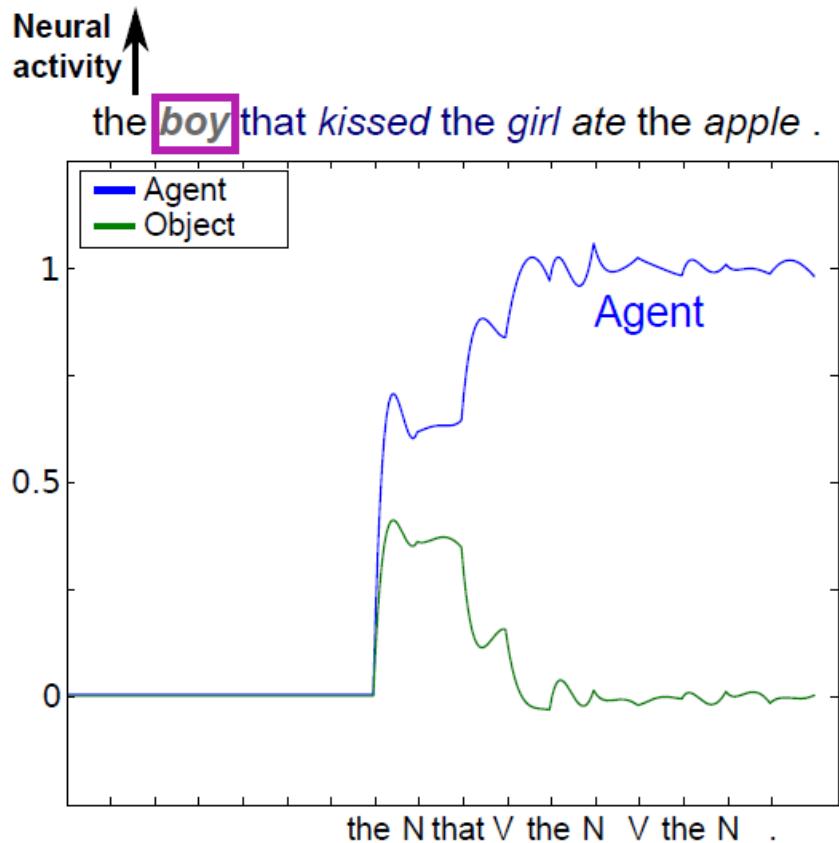


# Phase de test



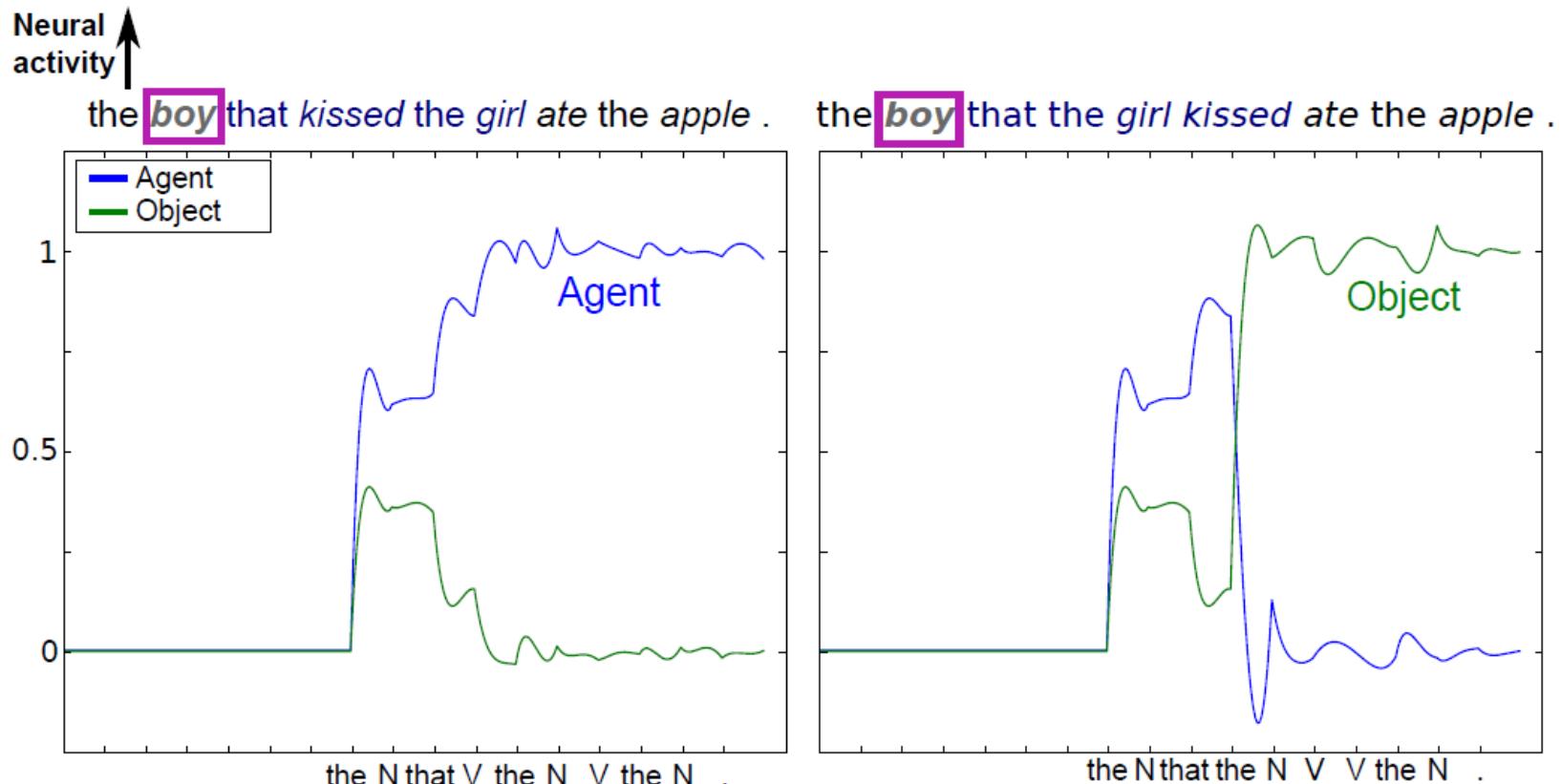
# More complex sentences

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



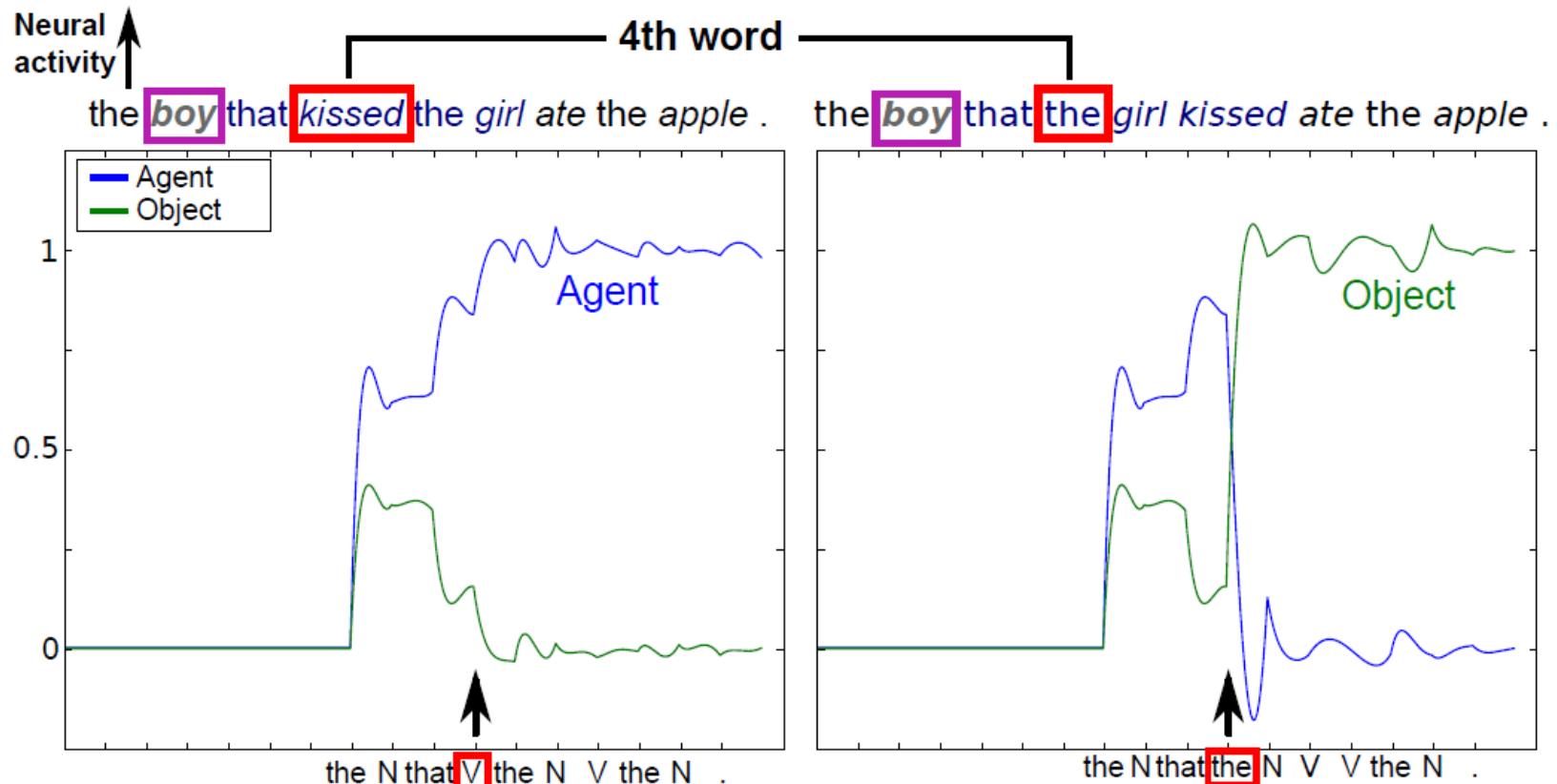
# More complex sentences

- 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)

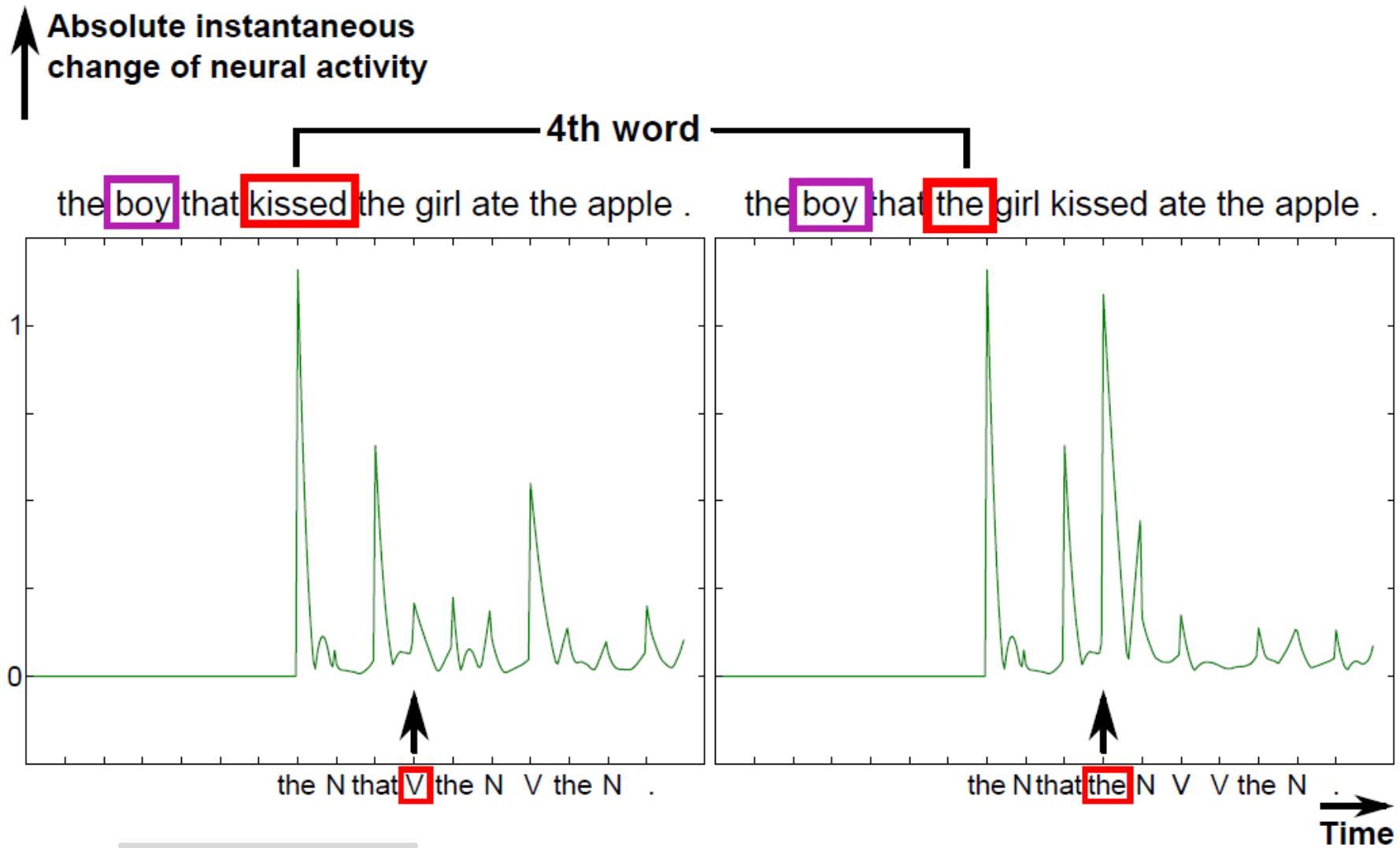


# More complex sentences

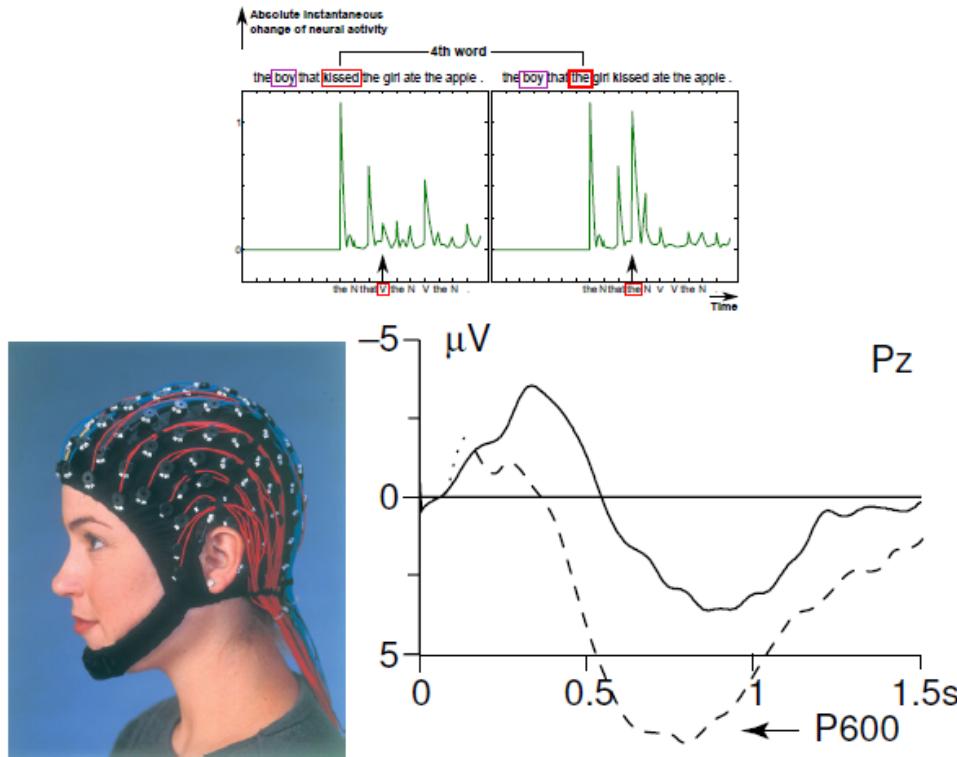
- On-going roles for the **relative clause** for the 1st SW: "boy"
  - The boy **that kissed** the girl ate the apple. (subject-relative)
  - The boy **that the girl kissed** ate the apple. (object-relative)



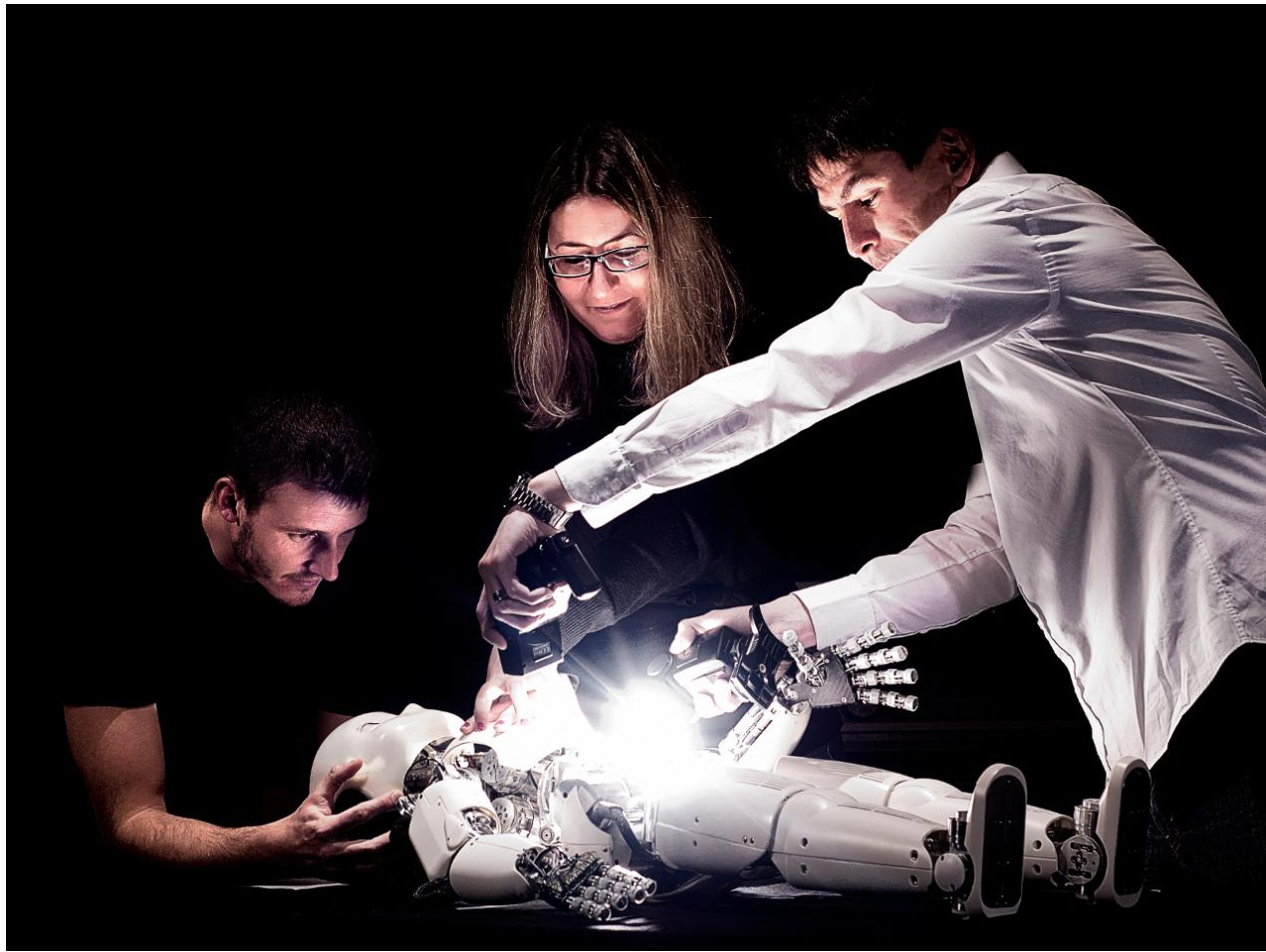
# More complex sentences



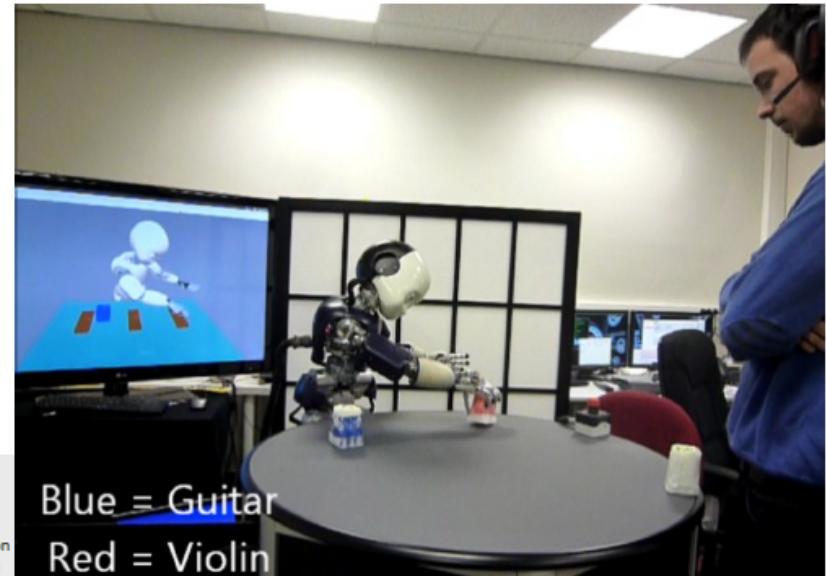
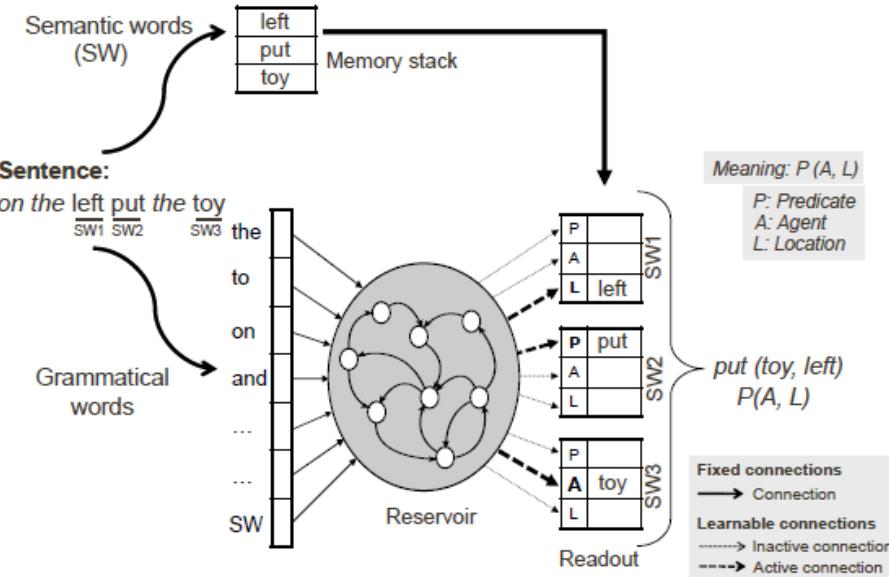
# More complex sentences



- 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



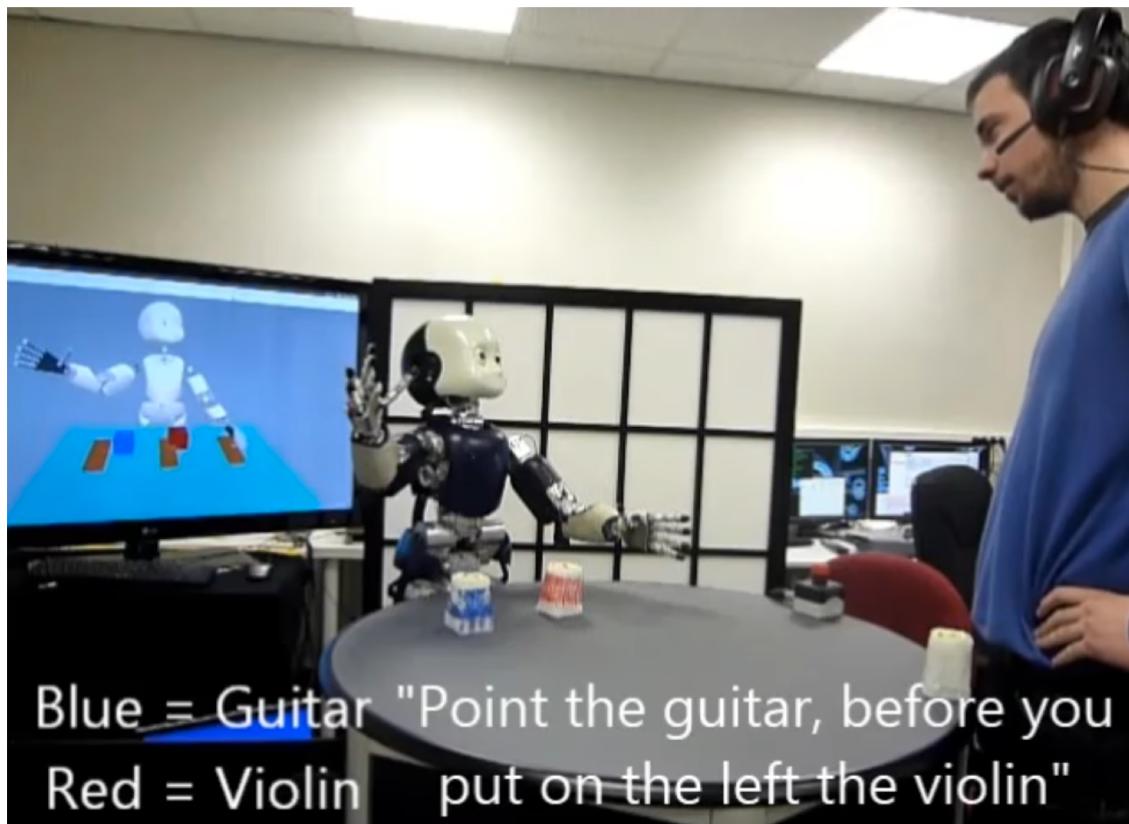
Hinaut et al. 2014

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

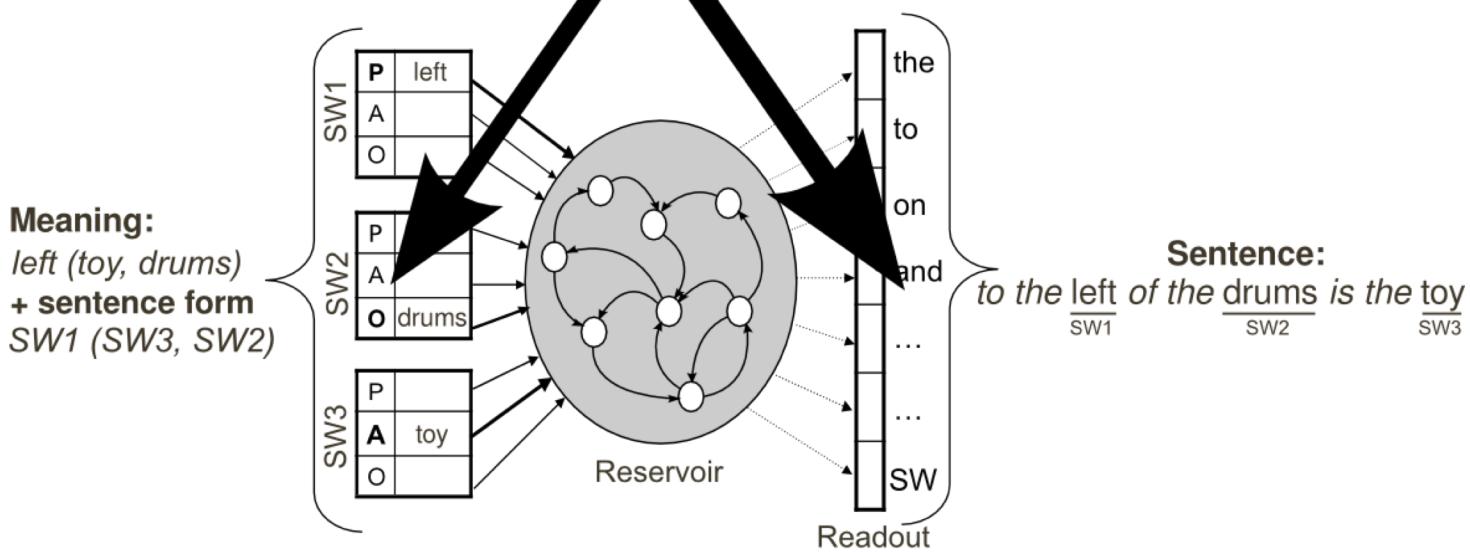
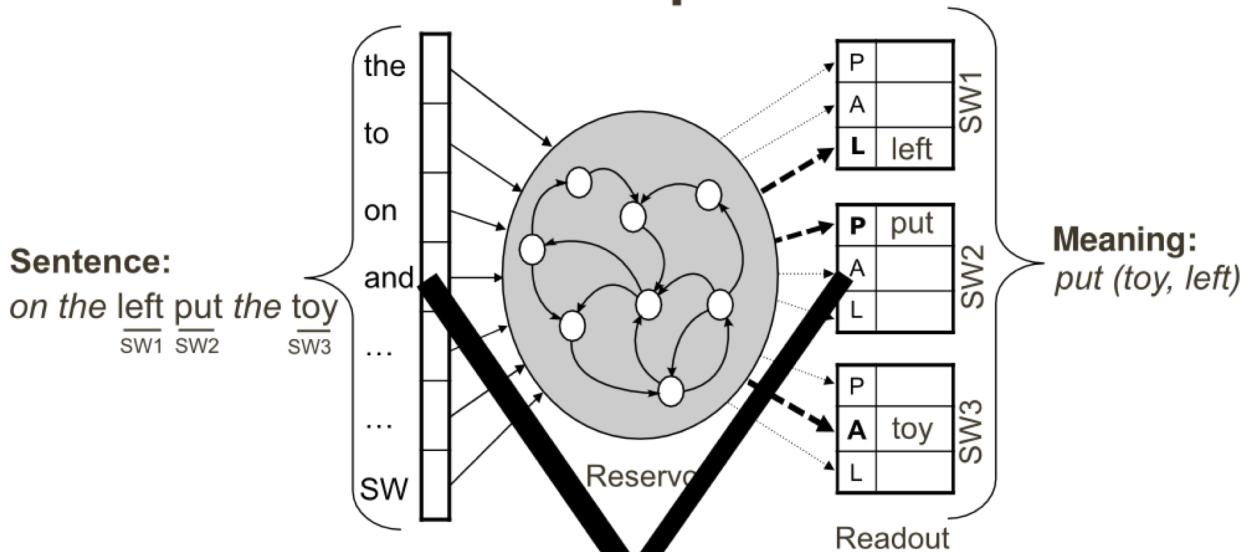


# Sentence Comprehension for HRI

<https://youtu.be/AUbJAupkU4M>



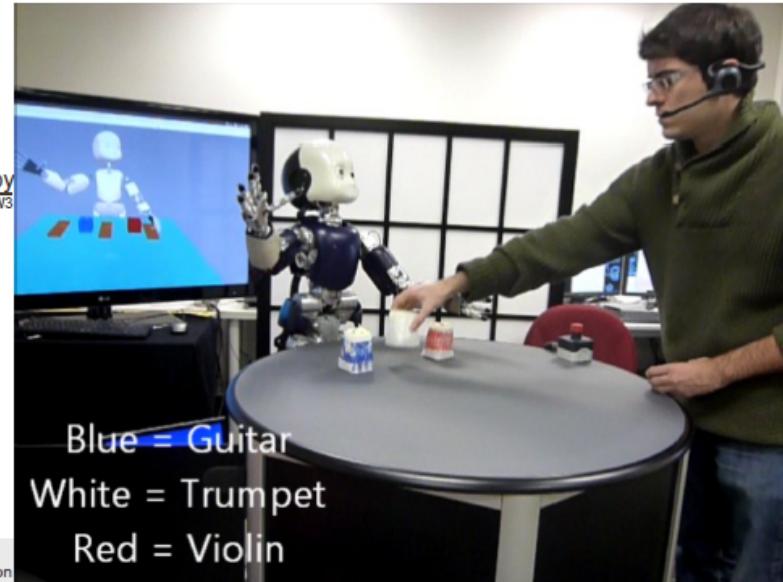
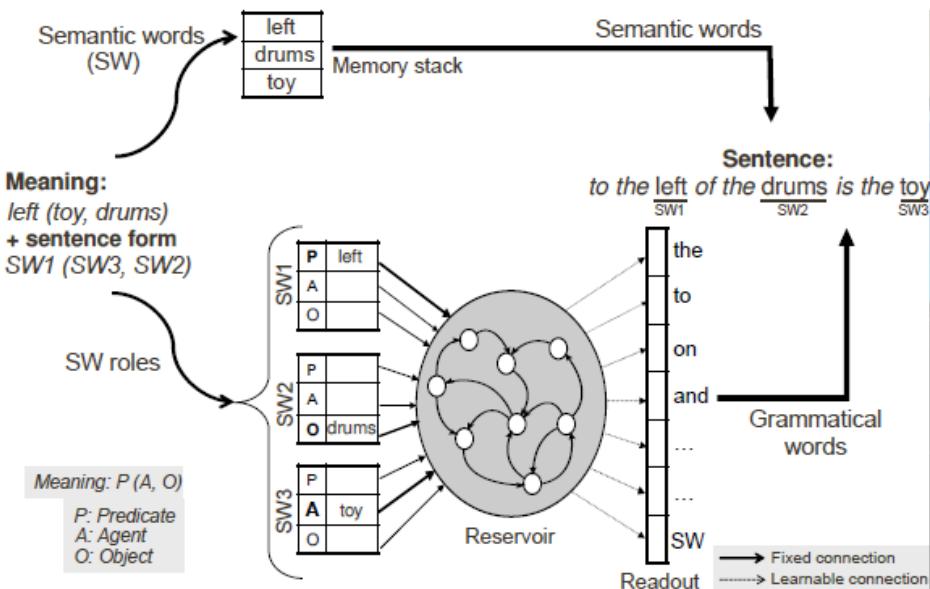
# Sentence comprehension model



# Sentence production ("inverse") model

Xavier HINAUT - Inria, Bordeaux, France

# Sentence Production



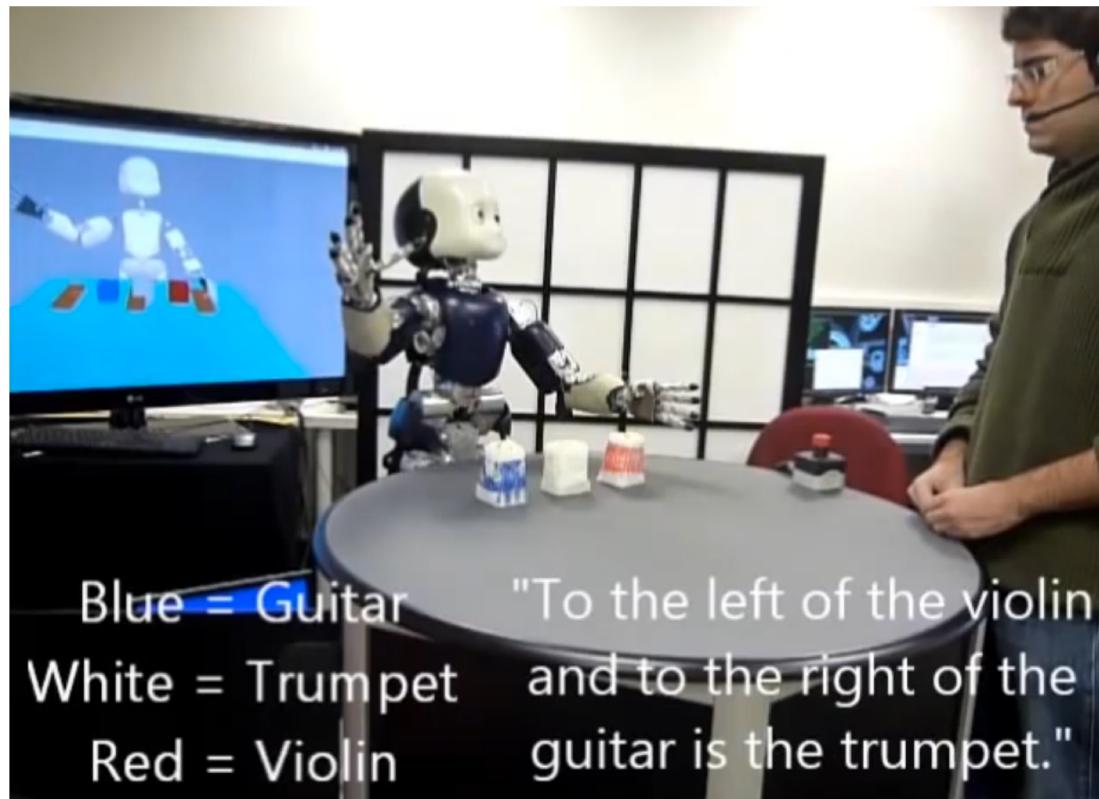
▶ Start movie

- 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

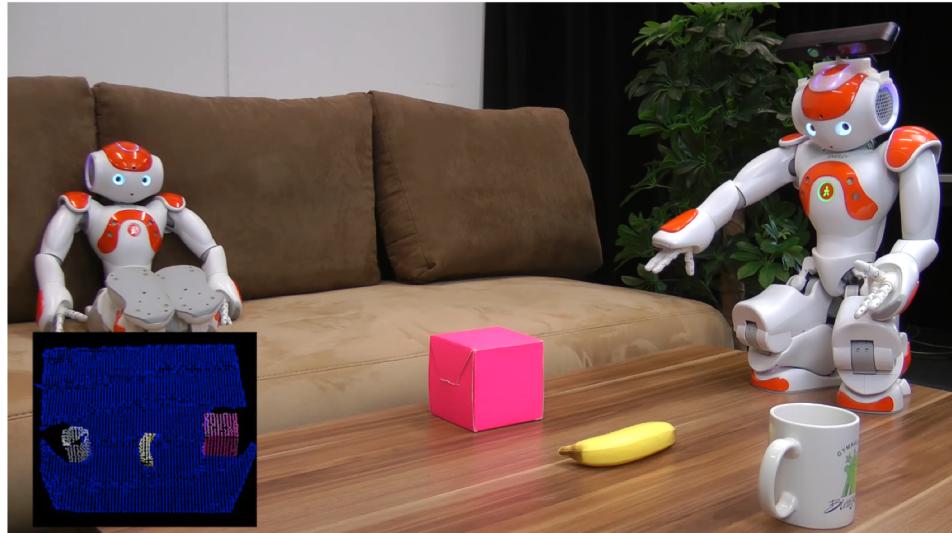
- point the triangle before grasping the circle
- put the cross to the left before grasping the circle
- point to the cross twice
- before you grasp the cross please grasp the triangle
- before pushing the triangle to the middle please push the cross to the right
- grasp the circle and then point to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

# Petit essai sur 15 langues !

- |   |                    |
|---|--------------------|
| 1) answer phone ; answer the phone                  | 1) English         |
| 2) geh an Telefon ; geh an das Telefon              | 2) German          |
| 3) contesta teléfono ; contesta el teléfono         | 3) Spanish         |
| 4) réponds téléphone ; réponds au téléphone         | 4) French          |
| 5) rispondi telefono ; rispondi al telefono         | 5) Italian         |
| 6) contesta telefon ; contesta el telefon           | 6) Catalan         |
| 7) 接电话;接那-通电话                                       | 7) Simpl. Mandarin |
| 8) 接電話;接那-通電話                                       | 8) Trad. Mandarin  |
| 9) jawab telefon ; jawab -kan panggilan telefon itu | 9) Malay           |
| 10) cevap ver telefon ; telefon -a cevap ver        | 10) Turkish        |
| 11) uchal phone ; phone uchal                       | 11) Marathi        |
| 12) dho javaab phon ; phon ka javaab dho            | 12) Hindi          |
| 13) telefono erantzun ; telefono -a erantzun        | 13) Basque         |
| 14) javab telfon ; telfon ra javab bede             | 14) Persian        |
| 15) atenda telefone ; atenda o telefone             | 15) Portuguese     |
| 16) вдигни телефон ; вдигни телефон -а              | 16) Bulgarian      |

# “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

...

[github.com/neuronalX](https://github.com/neuronalX)



# Des questions?



Campus Paris Saclay

FONDATION DE COOPERATION SCIENTIFIQUE



RobotDoC

Robotics for Development of Cognition



inria

informatics mathematics

# Tutoriel officiel de spaCy

[course.spacy.io](https://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 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.