

A dynamic action scene featuring Optimus Prime, the leader of the Autobots, in his red and blue robot form. He is positioned in the center, holding a large blue and black cannon in his right hand, which is firing a bright blue and white energy blast. The background is a dark, cloudy sky filled with numerous red laser beams and streaks of light, suggesting a high-stakes battle. Debris and sparks are visible in the air, adding to the chaotic atmosphere. The overall color palette is dominated by the reds and blues of Optimus Prime, the dark blues of the sky, and the bright reds of the enemy fire.

Word Embeddings

Mohamed Abbas KONATE

Plan

- Langage humain et machine
- Tokenisation : Choisir son vocabulaire
- Encodage : Représenter le vocabulaire

But du chapitre : Comprendre et connaître les grandes manières de représenter des données séquentielles textuelles

Langage humain



Dans un monde de chiffres, de 0 et de 1,
Un langage naît, mais sans âme ni destin,
L'ordinateur calcule, il apprend et devine,
Les mots se transforment en motifs et en lignes.

Il suit des algorithmes, des chemins de données,
Cherche des sens cachés, des pensées
encodées,
Mais ce n'est qu'une danse de logique et d'efforts,
Une illusion fragile d'un esprit qui dort.

Et je te demande, machine sans fin,
Avec tous tes circuits, peux-tu comprendre enfin
?

Suite de mots et caractères formant un sens.

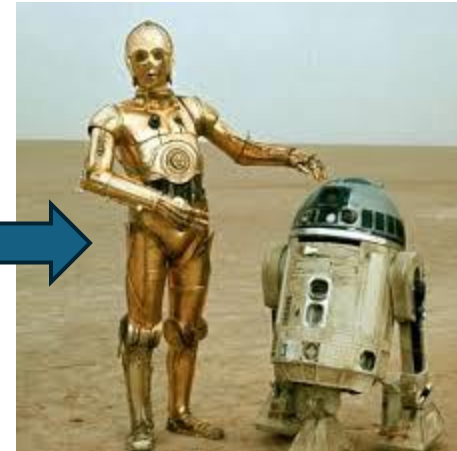
Langage humain et machine



Dans un monde de chiffres, de 0 et de 1,
Un langage naît, mais sans âme ni destin,
L'ordinateur calcule, il apprend et devine,
Les mots se transforment en motifs et en lignes.

Il suit des algorithmes, des chemins de données,
Cherche des sens cachés, des pensées
encodées,
Mais ce n'est qu'une danse de logique et d'efforts,
Une illusion fragile d'un esprit qui dort.

Et je te demande, machine sans fin,
Avec tous tes circuits, peux-tu comprendre enfin
?




Suite de mots et caractères formant un sens.

Opération sur des nombres

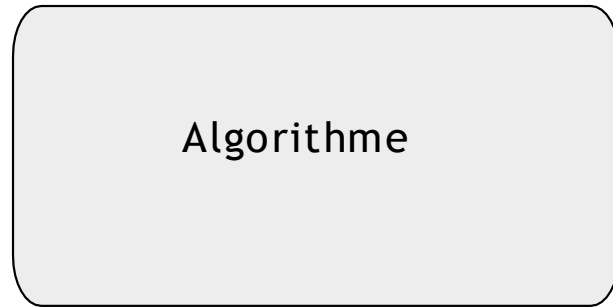
Langage Humain et machine

I saw a cat



Texte (entrée)

Langage Humain et machine



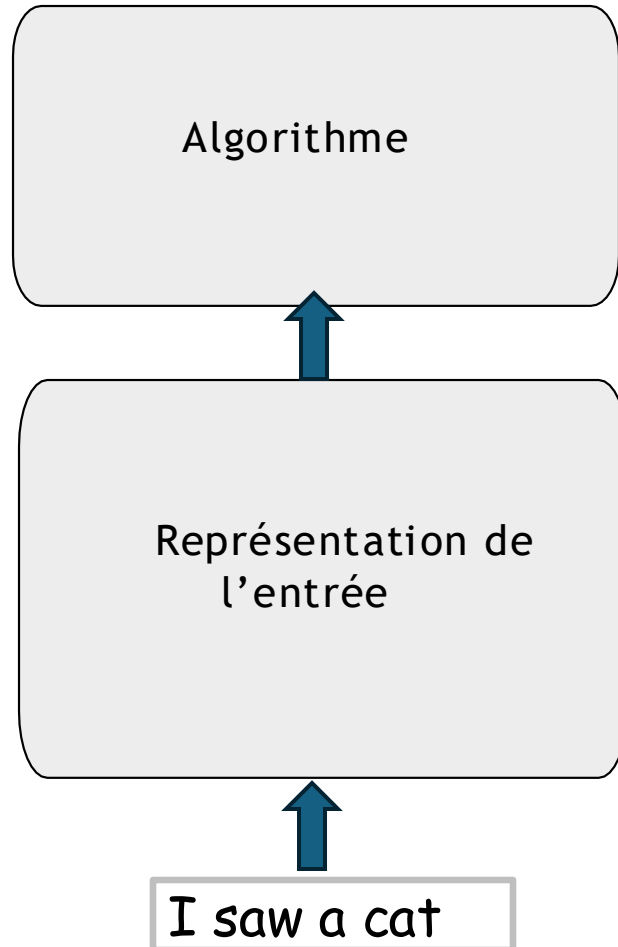
Algorithme

Tout algorithme permettant de résoudre une tâche de NLP (classification, traduction ...)

I saw a cat

Texte (entrée)

Langage Humain et machine



Tout algorithme permettant de résoudre une tâche de NLP

Représentation de l'entrée pour le modèle/algorithme

Texte (entrée)

Langage Humain et machine

- Comment représenter le langage humain?

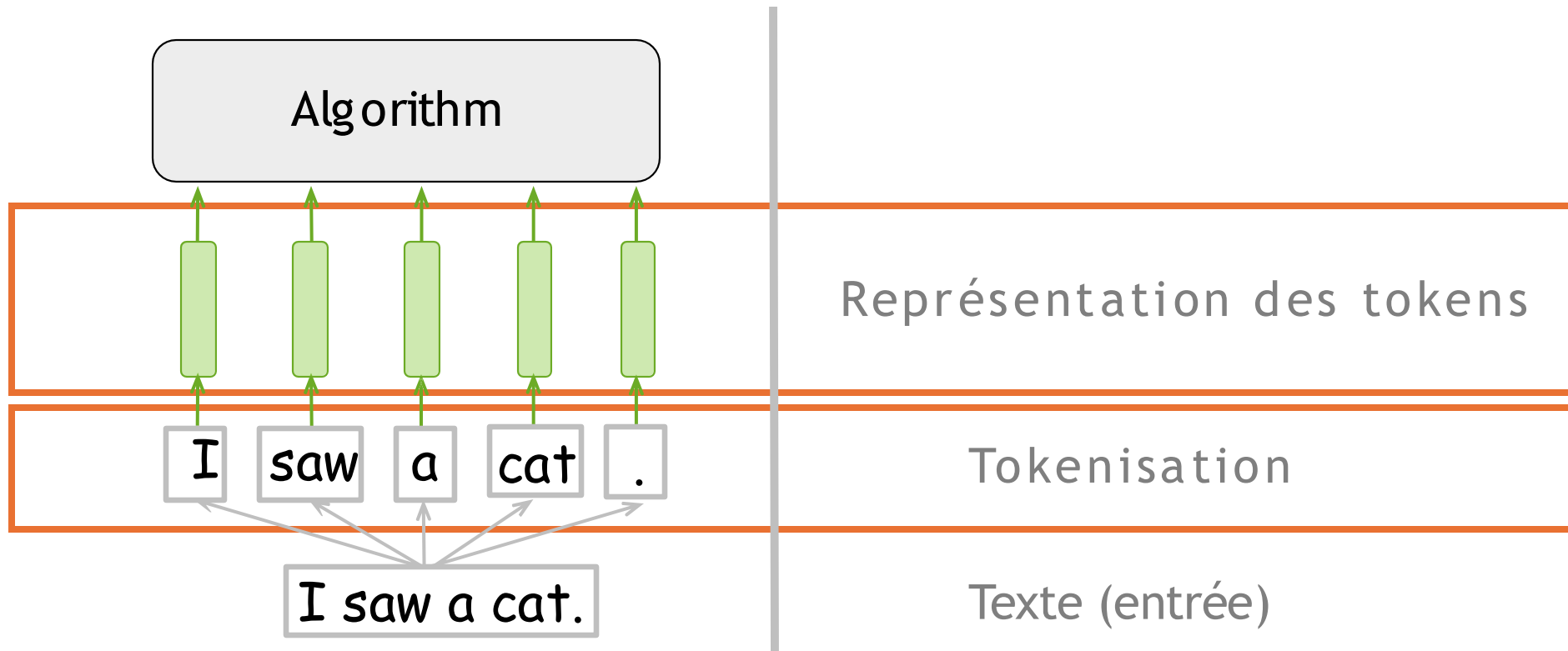
Langage Humain et machine

- Comment représenter le langage humain?
- Tokenisation

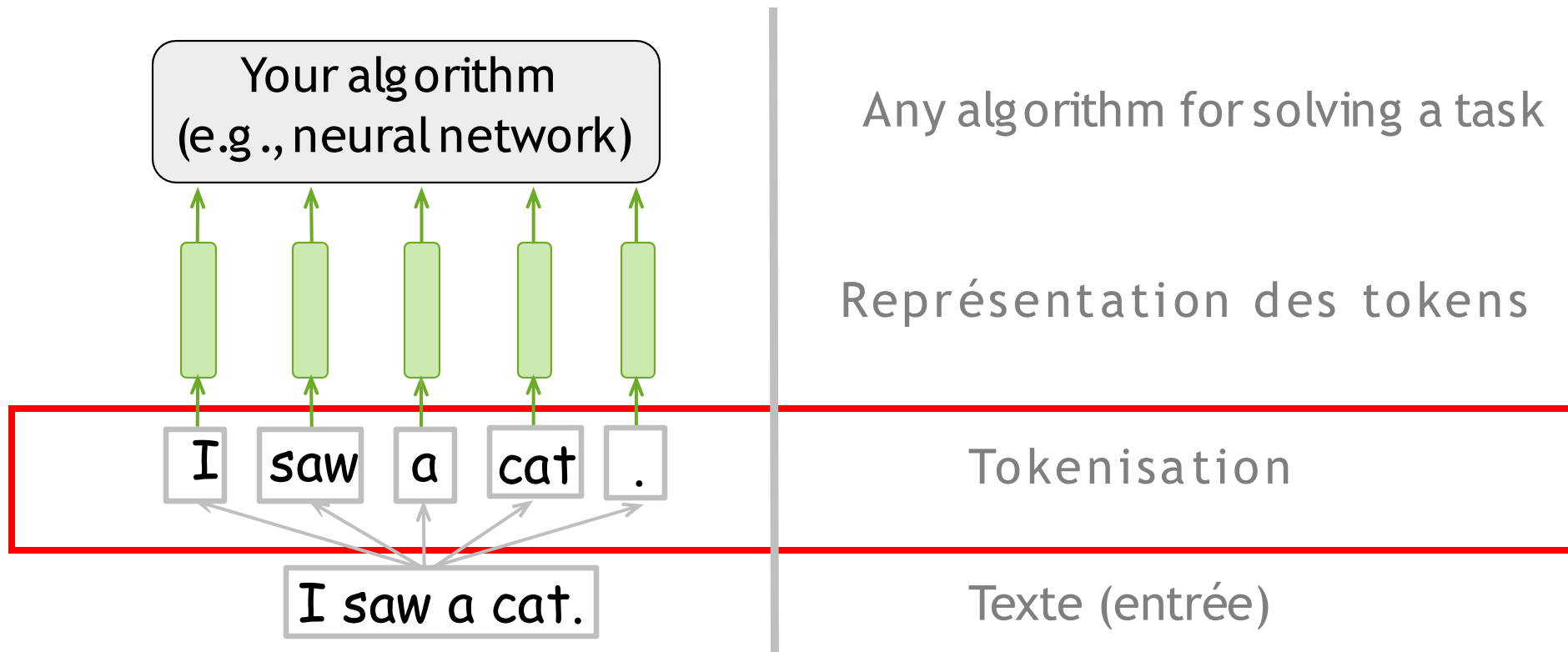
Langage Humain et machine

- Comment représenter le langage humain?
- Tokenisation
- Représentation du texte d'entrée : Embeddings

Langage Humain et machine



Tokenisation



Tokenisation

- Diviser la séquence d'entrée en petite parties : les « tokens »
- Élément atomique du langage pour la machine
- **Vocabulaire ou dictionnaire** : Ensemble de tous les tokens


Tokenisation

- Diviser la séquence d'entrée en petite parties : les « tokens »
- Élément atomique du langage pour la machine
- **Vocabulaire ou dictionnaire** : Ensemble de tous les tokens

Idée N°1 : Tokeniser les caractères

- Vocabulaire : Set des caractères individuel dans le corpus de texte


- Exemple :

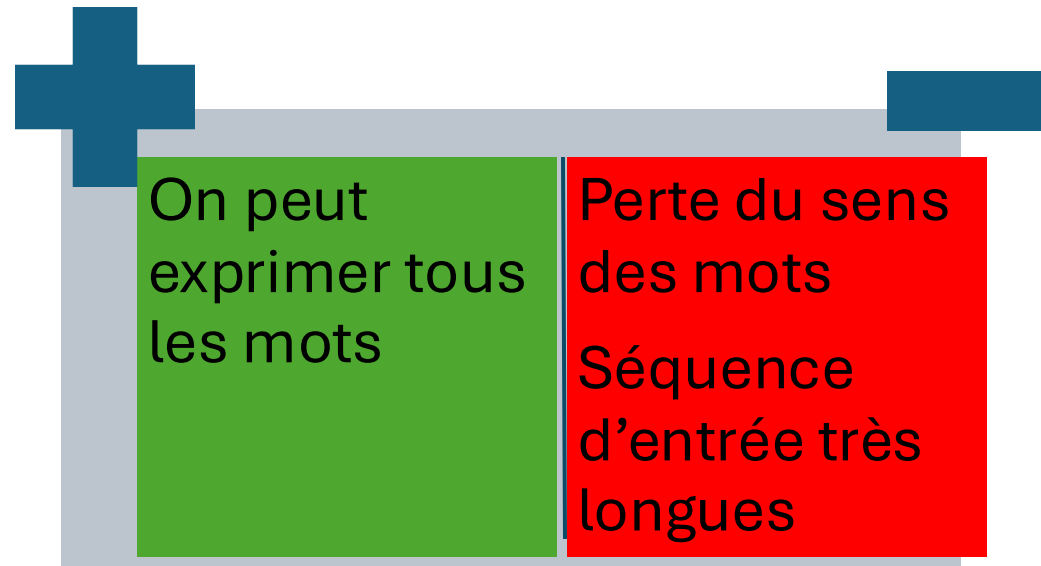
 = *[l, s, a, w, a, c, a, t, .]*

Idée N°1 : Tokeniser les caractères

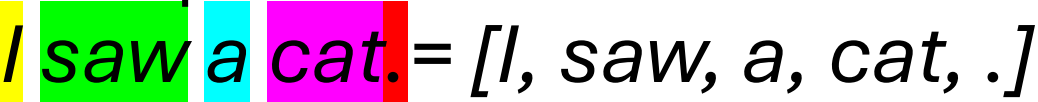
- Vocabulaire : Set des caractères individuel dans le corpus de texte

- Exemple :

 = $[l, s, a, w, a, c, a, t, .]$



Idée N°2 : Tokeniser les mots

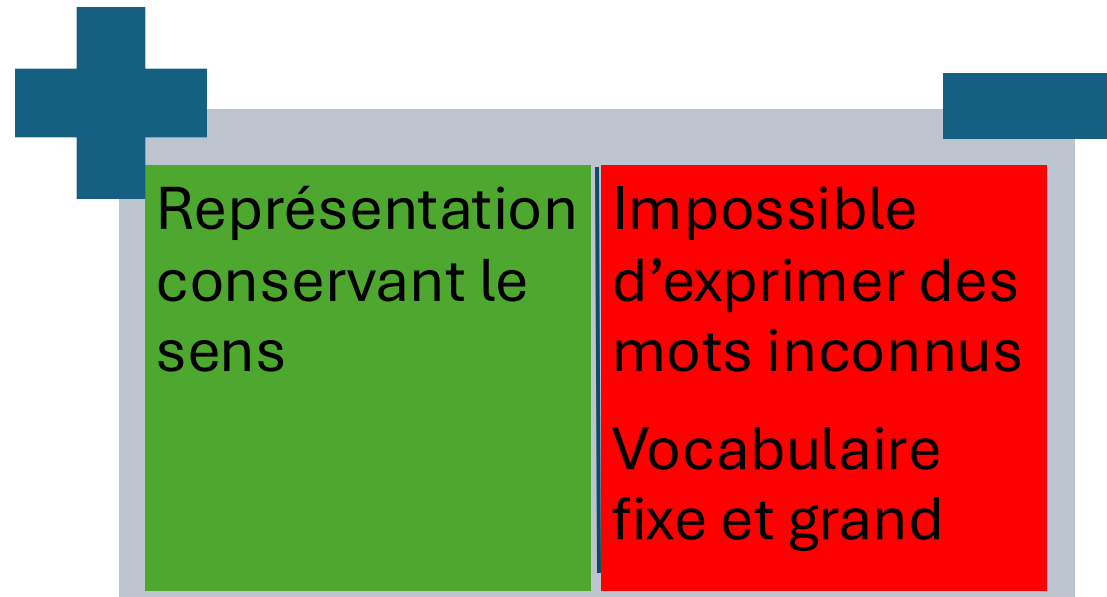
- Vocabulaire : Set des caractères individuel dans le corpus de texte
- Exemple :
= [I, saw, a, cat, .]

Idée N°2 : Tokeniser les mots

- Vocabulaire : Set des caractères individuel dans le corpus de texte

- Exemple :

I saw a cat. = *[I, saw, a, cat, .]*



Indexation de chaque token



Indexation de chaque token

Index Token dans le vocabulaire

39 1592 10 2548 5

I saw a cat .

1592



Vocabulary
size

Indexation de chaque token

Index Token dans le vocabulaire

39 1592 10 2548 5



I

saw

a

cat

.

10



Vocabulary
size

Indexation de chaque token

Index Token dans le vocabulaire

39 1592 10 2548 5



I

saw

a

cat

.

2548



Vocabulary
size

Indexation de chaque token

Index Token dans le vocabulaire

39 1592 10 2548 5



I



saw



a



cat



.

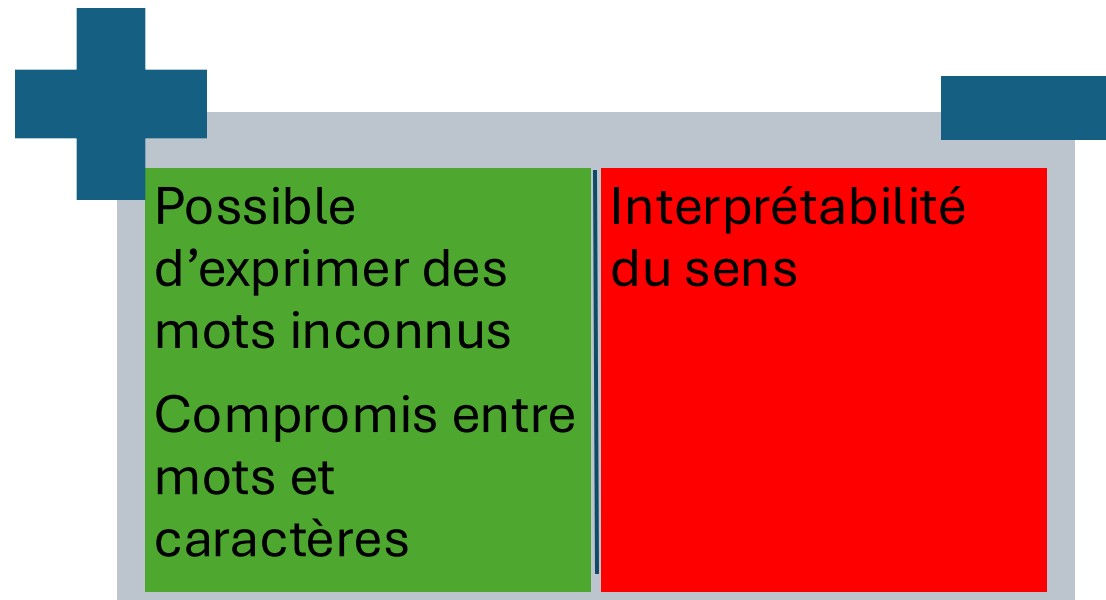
5



Vocabulary
size

Idée N° 3 : Tokeniser les sous-mots

- Se libérer de la contrainte du vocabulaire fixe
- Exprimer des mots absents du dataset initial



Byte Pair Encoding (BPE)



- Idée : **Apprendre à diviser** les mots en sous-mots
 - Exemple : “Manufacturing” = [“Manu”, “factur”, “ing”]
- L’algorithme de reference : Byte Pair Encoding

R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016), 2016. [Online]. Available: <https://arxiv.org/abs/1508.07909>

Byte Pair Encoding (BPE)

Comment ça marche :

Entrée : Un corpus de texte , Taille de vocabulaire souhaitée N

Sortie : Un vocabulaire de tokens (sous-mots)

1. Tokenisation en caractères:

Transformer chaque mot en caractères comme tokens initiaux.

Byte Pair Encoding (BPE)

Comment ça marche :

Entrée : Un corpus de texte , Taille de vocabulaire souhaitée N

Sortie : Un vocabulaire de tokens (sous-mots)

1. Tokenisation en caractères:

Transformer chaque mot en caractères comme tokens initiaux.

2. Operations de fusion:

1. Identifier la paire de jetons la plus fréquente.
2. Fusionner cette paire en un nouveau token.
3. Répétez l'opération jusqu'à ce que la taille de vocabulaire N souhaitée soit atteinte.

Byte Pair Encoding (BPE)

Comment ça marche :

Entrée : Un corpus de texte , Taille de vocabulaire souhaitée N

Sortie : Un vocabulaire de tokens (sous-mots)

1. Tokenisation en caractères:

Transformer chaque mot en caractères comme tokens initiaux.

2. Operations de fusion:

1. Identifier la paire de jetons la plus fréquente.
2. Fusionner cette paire en un nouveau token.
3. Répétez l'opération jusqu'à ce que la taille de vocabulaire N souhaitée soit atteinte.

3. Résultats:

1. Génère un vocabulaire de tokens (sous-mots) qui peut être utilisé pour la tokenisation du texte

Byte Pair Encoding (BPE)

- Texte d'entrée = [(cat,4),(mat,5),(mats,2),(mate,3),(ate,3),(eat,2)]
- Taille de vocabulaire voulu : 15
- Etape 1 : Vocabulaire actuel : [c,a,t,m,e,s]
 - Transformation en caractères [(c;a;t,4), (m;a;t,5), (m;a;t;s,2), (m;a;t;e,3), (a;t;e,3), (e;a;t,2)]
 - La paire la plus présente est : $(a;t,4+5+2+3+3+2) = 19$

Byte Pair Encoding (BPE)

- Texte d'entrée = [(cat,4),(mat,5),(mats,2),(mate,3),(ate,3),(eat,2)]
- Taille de vocabulaire final voulu : 15
- Etape 2 : Vocabulaire actuel : [c,a,t,m,e,s ,at]
 - Transformation en caractères
[(c;at,4),(m;at,5),(m;at;s,2),(m;at;e,3),(at;e,3), (e;at,2)]
 - La paire la plus présente est : (m;at, 5+2+3) = 10

Byte Pair Encoding (BPE)

- Texte d'entrée = [(cat,4),(mat,5),(mats,2),(mate,3),(ate,3),(eat,2)]
- Taille de vocabulaire final voulu : 15
- Etape 2 : Vocabulaire actuel : [c,a,t,m,e,s ;at, mat]
 - Transformation en caractères [(c;at,4),(~~mat~~,5),(mat;s,2),(mat;e,3),(at;e,3),(e;at,2)]
 - La **paire** la plus présente est : (c;at, 5) = 5

Byte Pair Encoding (BPE)

- Texte d'entrée = [(cat,4),(mat,5),(mats,2),(mate,3),(ate,3),(eat,2)]
- Taille de vocabulaire final voulu : 15
- Etape 2 : Vocabulaire actuel : [c,a,t,m,e,s ,at, mat, cat]
 - Transformation en caractères [~~(cat,4)~~,~~(mat,5)~~, (mat;s,2), (mat;e,3), (at;e,3), (e;at,2)]
 - La **paire** la plus présente est : (mat;e, 3) = 3

Byte Pair Encoding (BPE)

- Texte d'entrée = [(cat,4),(mat,5),(mats,2),(mate,3),(ate,3),(eat,2)]
- Taille de vocabulaire final voulu : 15
- Etape 2 : Vocabulaire actuel : [c,a,t,m,e,s ,at, mat, cat, mate]
 - Transformation en caractères [~~(cat,4)~~,~~(mat,5)~~, (mat;s,2),~~(mate,3)~~, (at;e,3), (e;at,2)]
 - La **paire** la plus présente est : (at;e, 3) = 3

Byte Pair Encoding (BPE)

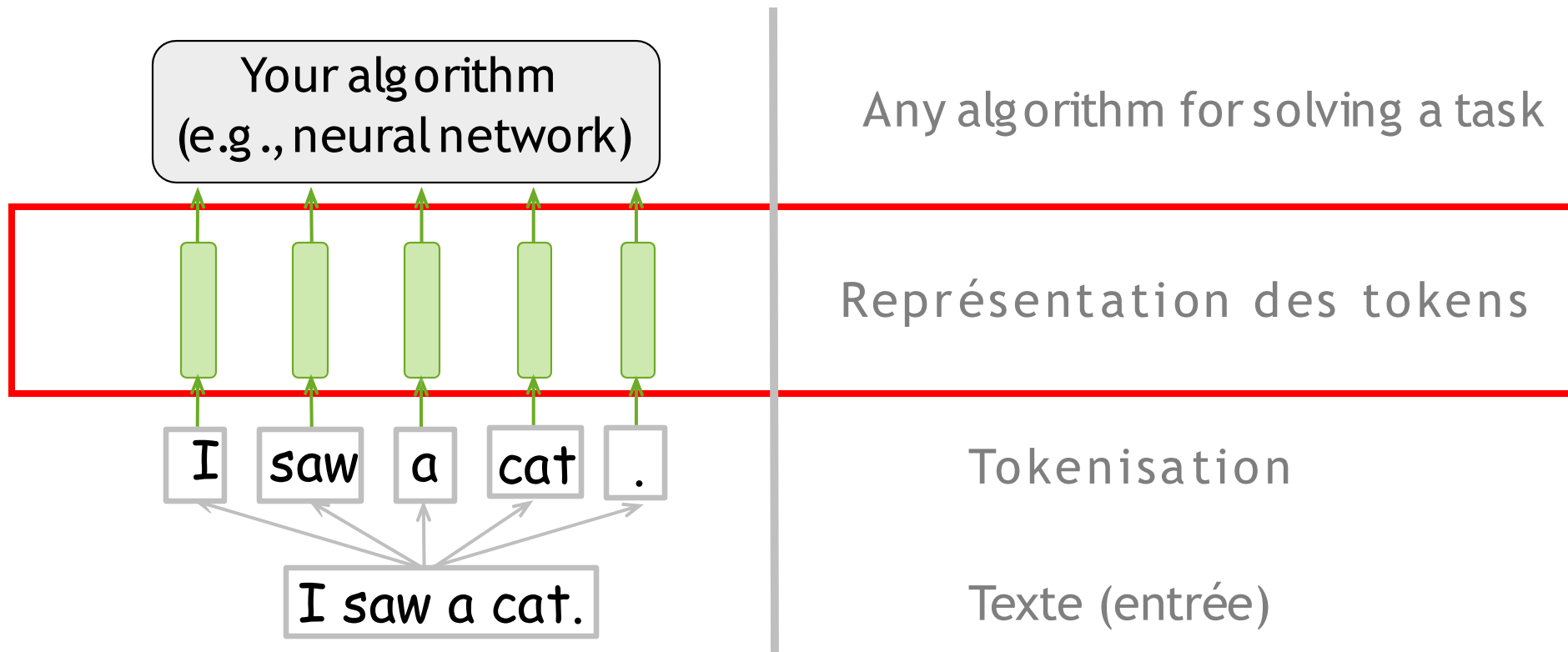
- Texte d'entrée = [(cat,4),(mat,5),(mats,2),(mate,3),(ate,3),(eat,2)]
- Taille de vocabulaire final voulu : 15
- Etape 2 : Vocabulaire actuel : [c,a,t,m,e,s ,at,mat,cat,mate,ate,mats]
 - Transformation en caractères [~~(cat,4)~~,~~(mat,5)~~,~~(mat;s,2)~~,~~(mate,3)~~,~~(ate,3)~~,~~(e;at,2)~~]
 - Fin

Exercice

- Exercice : Coder sans librairie externe le tokeniseur de GPT2.



Apprentissage de representation



Apprentissage de representation

- Maintenant qu'on a notre vocabulaire : représenter le langage pour la machine
- Représentation continue

Apprentissage de representation

- Maintenant qu'on a notre vocabulaire : représenter le langage pour la machine
- Représentation continue
- Réduction de la dimensionnalité

Apprentissage de representation

- Maintenant qu'on a notre vocabulaire : représenter le langage pour la machine
- Représentation continue
- Réduction de la dimensionnalité
- Apprentissage automatique

Représentation de mots via des vecteurs :

« One-hot »

- Question : Quelle est la façon la plus simple de représenter un mot (à partir d'une liste de N mots) avec un vecteur ?

Représentation de mots via des vecteurs :

« One-hot »

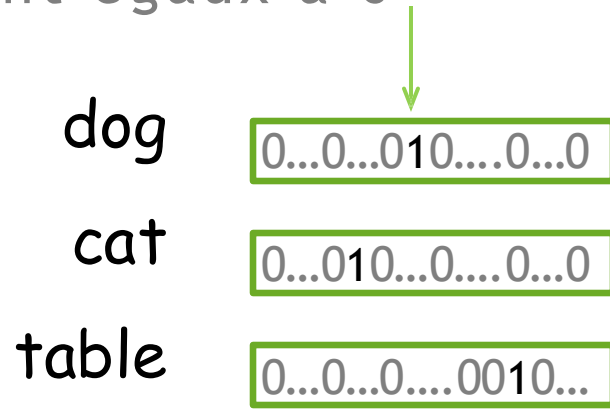
- **Question:** Quelle est la façon la plus simple de représenter un mot (à partir d'une liste de N mots) avec un vecteur ?
- Voici une idée :
- Créez des vecteurs à N dimensions pour chaque mot
- Sauf l'élément correspondant à 1, et zéro le reste

C'est ce qu'on appelle "one-hot" vectors

Représentation de mots via des vecteurs :

« One-hot »

L'un est égal à 1, les autres
sont égaux à 0



The diagram illustrates one-hot vectors for three words: 'dog', 'cat', and 'table'. Each word is followed by a horizontal box representing a vector of binary values (0s and 1s). A green arrow points from the text 'L'un est égal à 1, les autres sont égaux à 0' to the first '1' in the 'dog' vector. The vectors are: 'dog' (0...0...010...0...0), 'cat' (0...010...0...0...0), and 'table' (0...0...0...0010...).

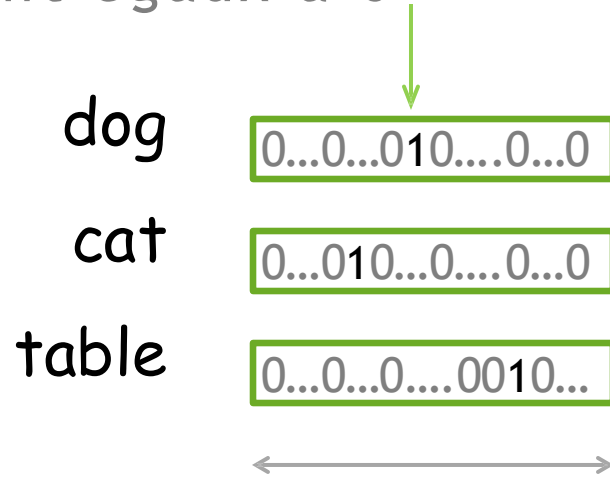
dog	0...0...010...0...0
cat	0...010...0...0...0
table	0...0...0...0010...

←————→
Embedding dimension = Taille du vocabulaire

Représentation de mots via des vecteurs :

« One-hot vectors »

L'un est égal à 1, les autres
sont égaux à 0




Des
problèmes ?

Embedding dimension = Taille du vocabulaire

Limitations des One-hot Vectors

L'un est égal à 1, les autres sont égaux à 0



motel	0...0...010...0...0
chat	0...010...0...0...0
hotel	0...0...0...0010...

Problèmes:

- La taille du vecteur est trop grande
- Les vecteurs ne savent rien du sens des mots, par exemple, **chat** est aussi proche que **motel** comme il est à **hotel** !

Exemple : dans une recherche sur le web, si un utilisateur effectue une recherche pour « *motel* », nous aimerions correspondre aux documents contenant « *hôtel* ».

Il n'y a pas de notion naturelle de similitude pour les vecteurs one-hot !

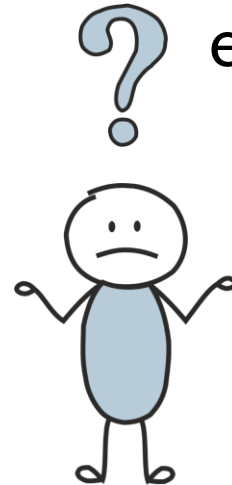
Limitations of One-hot Vectors

L'un est égal à 1, les autres sont égaux à 0

motel	0...0...010...0...0
chat	0...010...0...0...0
hôtel	0...0...0...0010...

Problèmes:

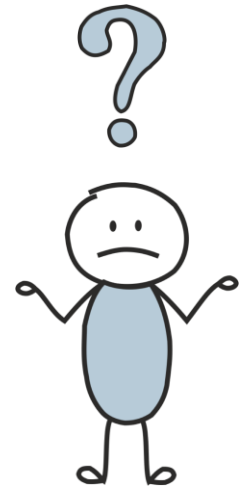
- La taille du vecteur est trop grande
- Les vecteurs ne savent rien du sens des mots, par exemple, **chat** est aussi proche que **motel** comme il est à **hôtel** !



Qu'est-ce que le sens ?

Qu'est-ce que le sens ?

Savez-vous ce que signifie le mot **bandji** ?



Qu'est-ce que le sens ?

Regardez maintenant comment ce mot est utilisé dans différents contextes :

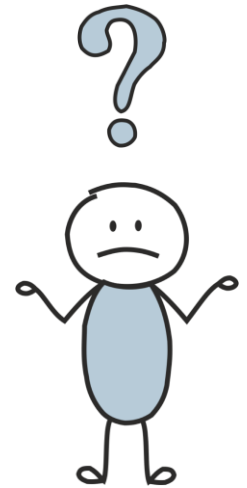
Une bouteille de **bandji** est sur la table.

Tout le monde aime le **bandji**.

Le **bandji** vous rend ivre.

Nous fabriquons le **bandji** avec de la sève de palmier.

Que signifie **bandji** ?



Qu'est-ce que le sens ?

Now look how this word is used in different contexts:

Une bouteille de **bandji** est sur la table.

Tout le monde aime le **bandji**.

Le **bandji** vous rend ivre.

Nous fabriquons le **bandji** avec de la sève de palmier.

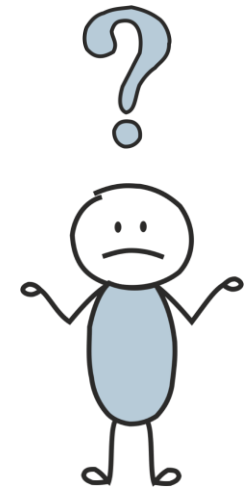


Le bandji est une sorte de
boisson alcoolisée à base de
sève de palmier.



Avec le **contexte**, vous pouvez comprendre le **sens** !

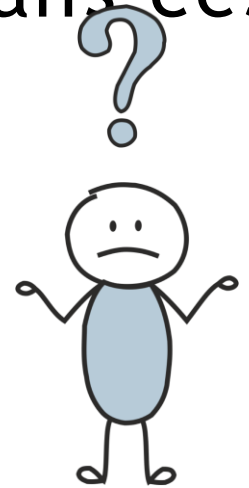
Comment avez-vous fait cela ?



Qu'est-ce que le sens ?

- (1) Une bouteille de _____ est sur la table.
- (2) Tout le monde aime le _____.
- (3) Le _____ vous rend ivre.
- (4) Nous fabriquons le _____ avec de la sève de palmier.

Quels autres mots
s'inscrivent dans ces
contextes ?



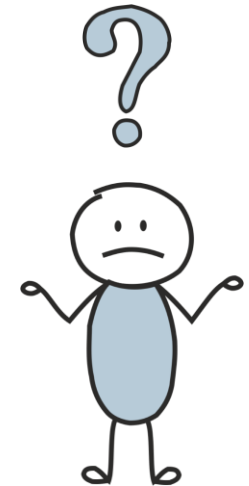
Qu'est-ce que le sens ?

- (1) Une bouteille de _____ est sur la table.
- (2) Tout le monde aime le _____.
- (3) Le _____ vous rend ivre.
- (4) Nous fabriquons le _____ avec de la sève de palmier.

Quels autres mots
s'inscrivent dans
ces contextes ?

	(1)	(2)	(3)	(4)	...	← Contextes
bandji	1	1	1	1		
heureux	0	0	0	0		
tournevis	0	1	0	0		
chat	0	1	0	0		
vodka	1	1	1	0		

← Les lignes affichent des propriétés contextuelles : 1 si un mot peut apparaître dans le contexte, 0 si ce n'est pas le cas



Qu'est-ce que le sens ?

(1) A bottle of _____ is on the table.

(2) Everyone likes _____ .

(3) _____ makes you drunk.

(4) We make _____ out of corn.

	(1)	(2)	(3)	(4)	...
bandji	1	1	1	1	
heureux	0	0	0	0	
tournevis	0	1	0	0	
chat	0	1	0	0	
vodka	1	1	1	0	

les rangées sont similaires

Qu'est-ce que le sens ?

(1) A bottle of _____ is on the table.

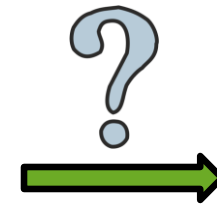
(2) Everyone likes _____ .

(3) _____ makes you drunk.

(4) We make _____ out of corn.

	(1)	(2)	(3)	(4)	...
bandji	1	1	1	1	
heureux	0	0	0	0	
tournevis	0	1	0	0	
chat	0	1	0	0	
vodka	1	1	1	0	

les rangées sont
similaires



Les significations
des mots sont
similaires

Hypothèse distributive

- *Des mots similaires se produisent dans des contextes similaires*

“You shall know a word by the company it keeps” (John Rupert Firth 1957)



- Base de nombreux modèles NLP modernes.

As an establishment providing accommodations **hotel** provide a variety of amenities ...
A motel, an abbreviation for "motor **hotel**", is a small-sized low-rise lodging ...
One of the first **hotel** was opened in Exeter in 1768 ...

Ces mots de contexte représenteront « hôtel »

Hypothèse distributionnelle

Des mots qui apparaissent fréquemment dans des contextes similaires ont une signification similaire.

Idée principale :

Nous devons mettre des informations sur les contextes dans des vecteurs de mots.

Représentation de mots via des vecteurs : « Word2Vec »

- Apprendre la représentation vectorielle de telle sorte que les mots de signification similaire soient plus proches.

$$\begin{array}{cc} \begin{array}{c} \text{“motel”} \\ \rightarrow \end{array} & \begin{bmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{bmatrix} & \begin{array}{c} \text{“hotel”} \\ \rightarrow \end{array} & \begin{bmatrix} 0.413 \\ 0.582 \\ -0.007 \\ 0.247 \\ 0.216 \\ -0.718 \\ 0.147 \\ 0.051 \end{bmatrix} \end{array}$$

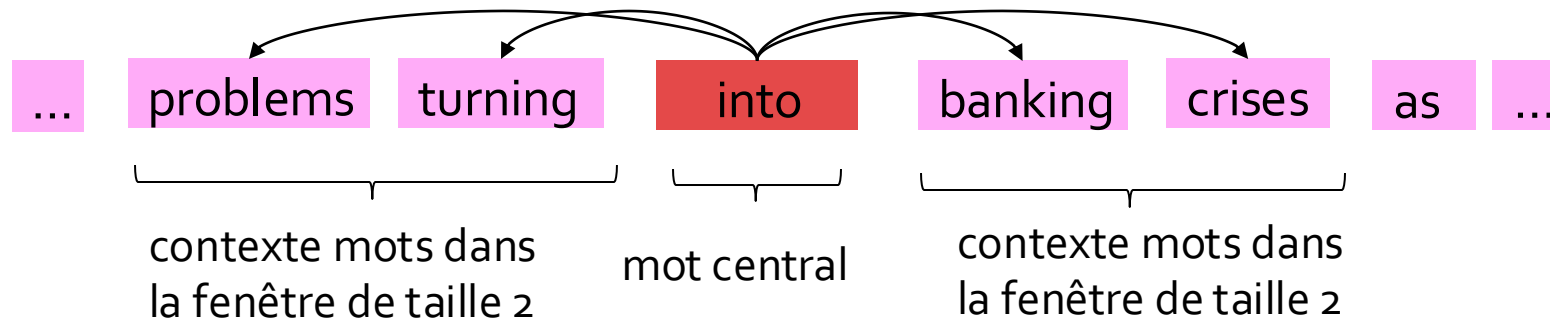
Word vectors sont aussi appelées (word) embeddings or (neural) word representations

Word2Vec : Vue d'ensemble

- Word2vec [Mikolov et al. 2013] est un cadre pour l'apprentissage des vecteurs de mots
- Collecter un large corpus de phrases (par exemple, Wikipédia)
- Chaque mot d'un vocabulaire fixe est représenté par un vecteur

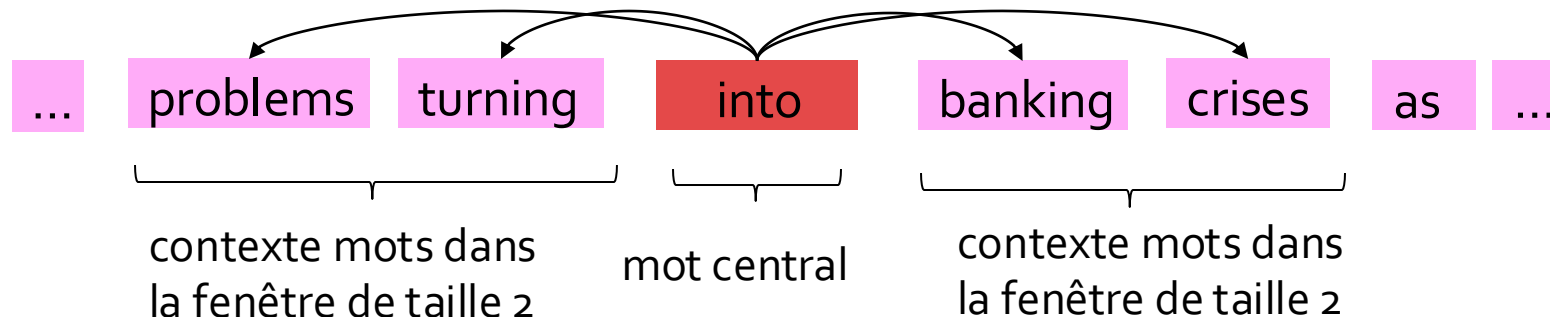
Word2Vec : Vue d'ensemble

- Word2vec [Mikolov et al. 2013] est un cadre pour l'apprentissage des vecteurs de mots
- Collecter un large corpus de phrases (par exemple, Wikipédia)
- Chaque mot d'un vocabulaire fixe est représenté par un vecteur



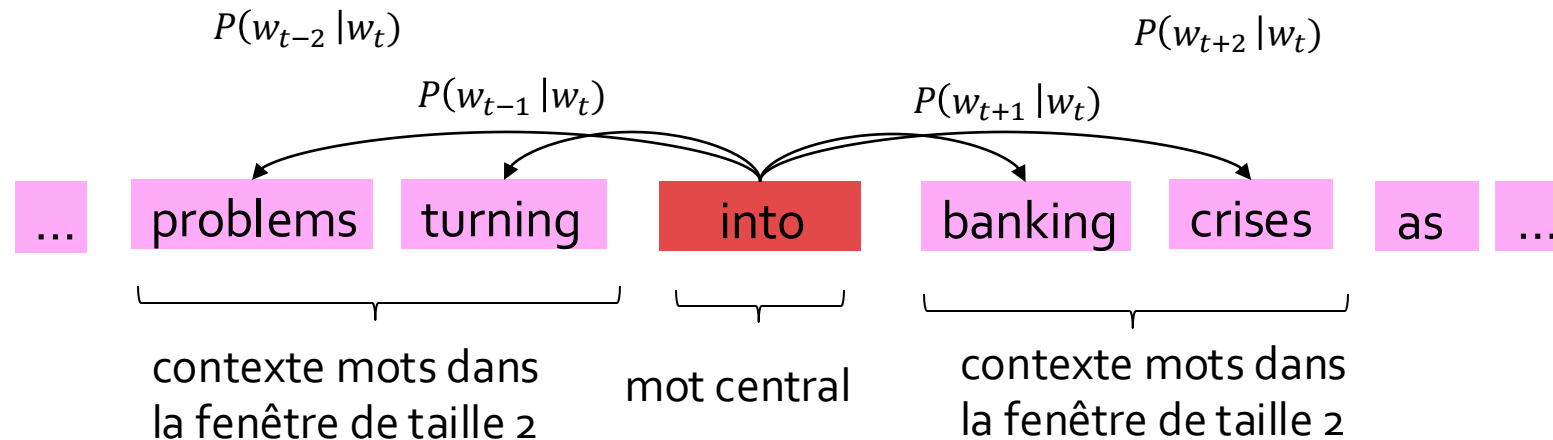
Word2Vec : Vue d'ensemble

- Word2vec [Mikolov et al. 2013] est un cadre pour l'apprentissage des vecteurs de mots
- Collecter un large corpus de phrases (par exemple, Wikipédia)
- Chaque mot d'un vocabulaire fixe est représenté par un vecteur
- Pour chaque position dans le texte, considérez le mot « central » c et les mots contextuels (« à l'extérieur ») o



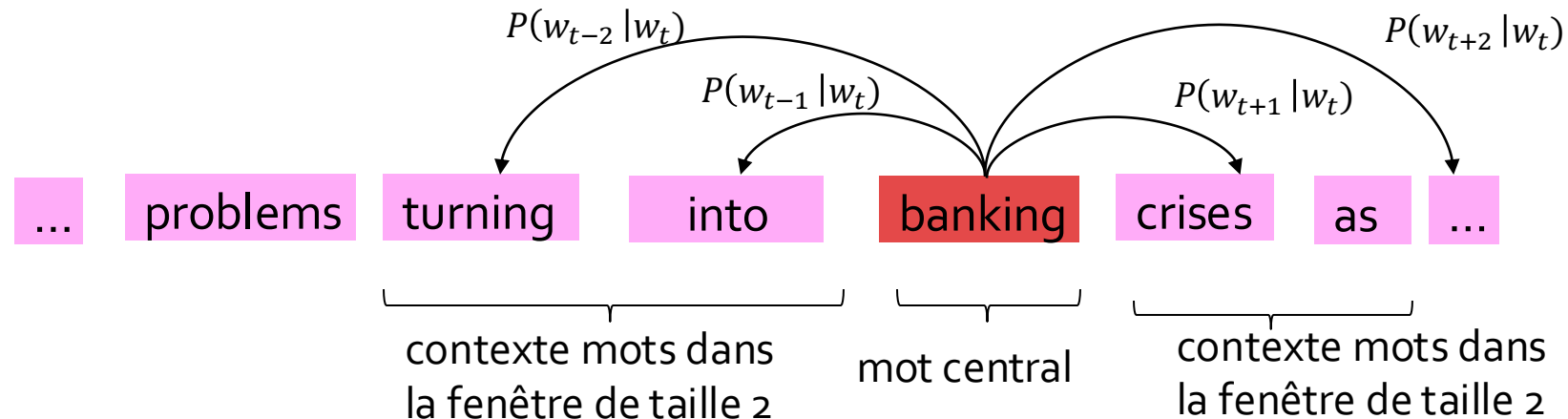
Word2Vec : Vue d'ensemble

- Word2vec [Mikolov et al. 2013] est un cadre pour l'apprentissage des vecteurs de mots
- Collecter un large corpus de phrases (par exemple, Wikipédia)
- Chaque mot d'un vocabulaire fixe est représenté par un vecteur
- Pour chaque position dans le texte, considérez le mot « central » c et les mots contextuels (« à l'extérieur ») o
- Définir la probabilité de o étant donné c (ou vice versa) : $P(c|o)$



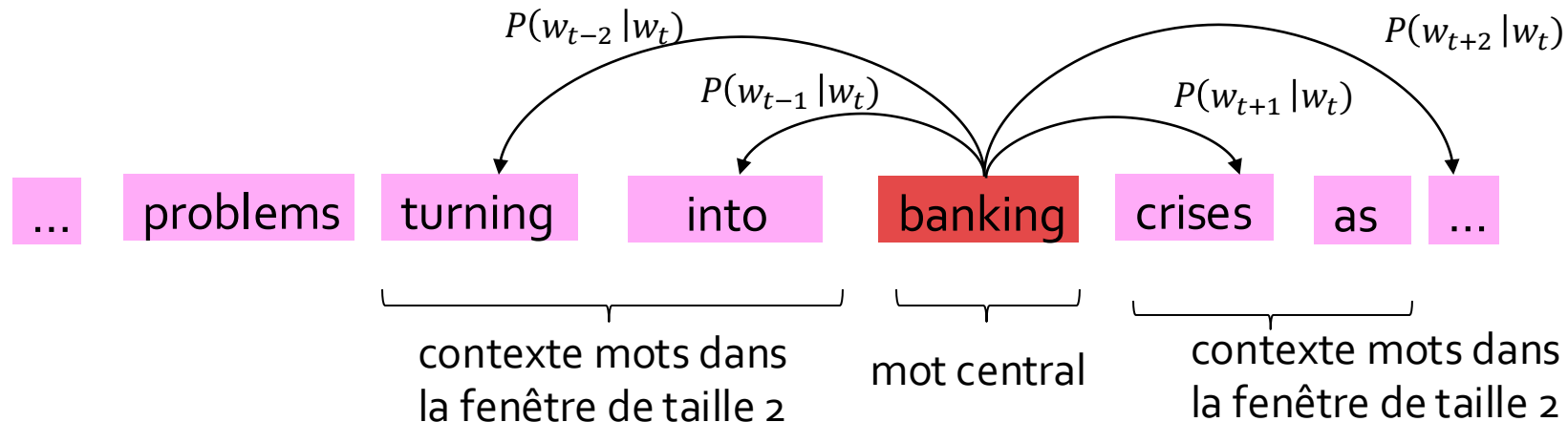
Word2Vec : Vue d'ensemble

- Word2vec [Mikolov et al. 2013] est un cadre pour l'apprentissage des vecteurs de mots
- Collecter un large corpus de phrases (par exemple, Wikipédia)
- Chaque mot d'un vocabulaire fixe est représenté par un vecteur
- Pour chaque position dans le texte, considérez le mot « central » c et les mots contextuels (« à l'extérieur ») o
- Définir la probabilité de o étant donné c (ou vice versa) : $P(c|o)$



Word2Vec : Vue d'ensemble

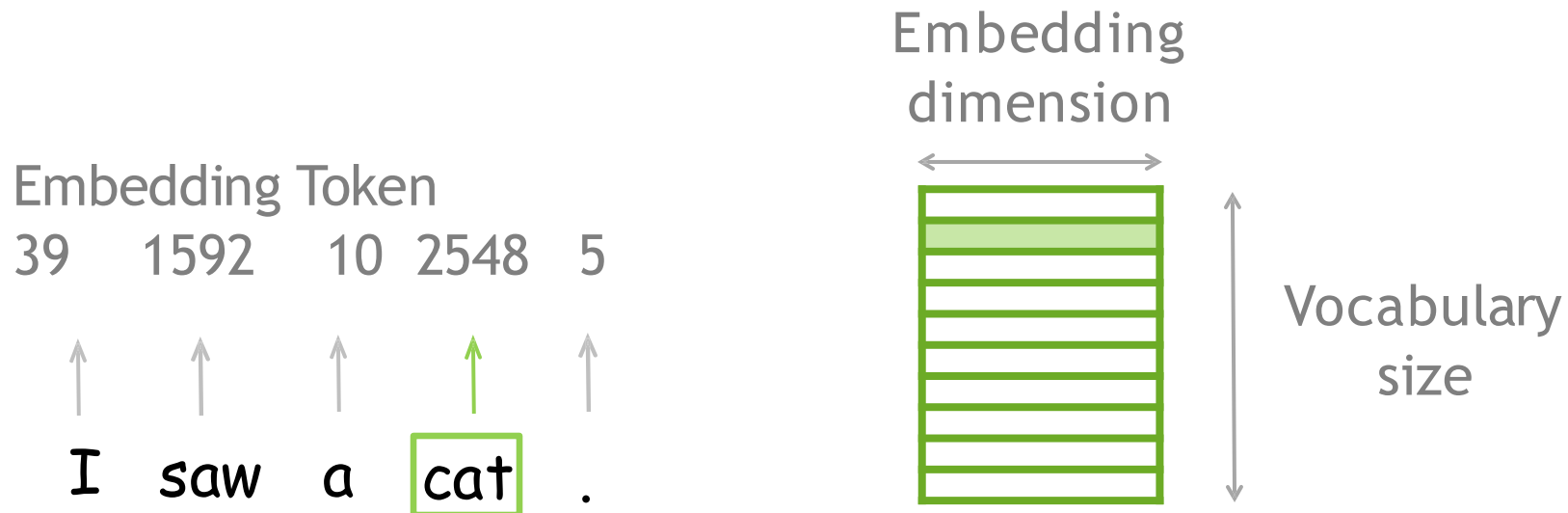
- Word2vec [Mikolov et al. 2013] est un cadre pour l'apprentissage des vecteurs de mots
- Collecter un large corpus de phrases (par exemple, Wikipédia)
- Chaque mot d'un vocabulaire fixe est représenté par un vecteur
- Pour chaque position dans le texte, considérez le mot « centre » c et les mots « hors contexte » o
- Définir la probabilité de o étant donné c (ou vice versa) : $P(c|o)$
- Optimiser les paramètres du modèle pour à ajuster les embeddings pour maximiser $P(c|o)$



Représentation de mots via des vecteurs : Apprentissage



- Apprendre la **représentation vectorielle directement** dans le processus d'entraînement.
- Apprendre une matrice E : Taille Vocabulaire * Taille Embedding



Représentation de mots via des vecteurs : Apprentissage

- Apprendre la **représentation vectorielle directement** dans le processus d'entraînement.
 - Apprendre une matrice E : Taille Vocabulaire * Taille Embedding
 - Par exemple : Open AI GPT2 : Taille Vocabulaire = 50247 subwords (Tokens BPE) ; Taille embedding : 768 , soit $E = 50247 * 768$ paramètres
- Où est la réduction de dimension ????

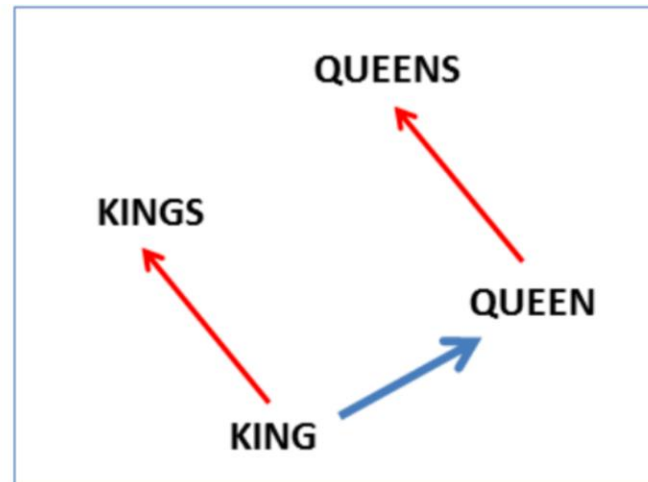
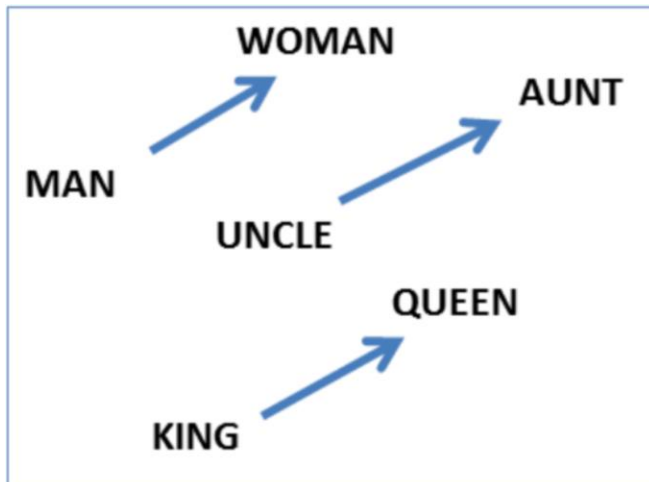


Analyse des embeddings

De nombreuses relations sémantiques et syntaxiques entre les mots sont (presque) linéaires dans l'espace des embeddings

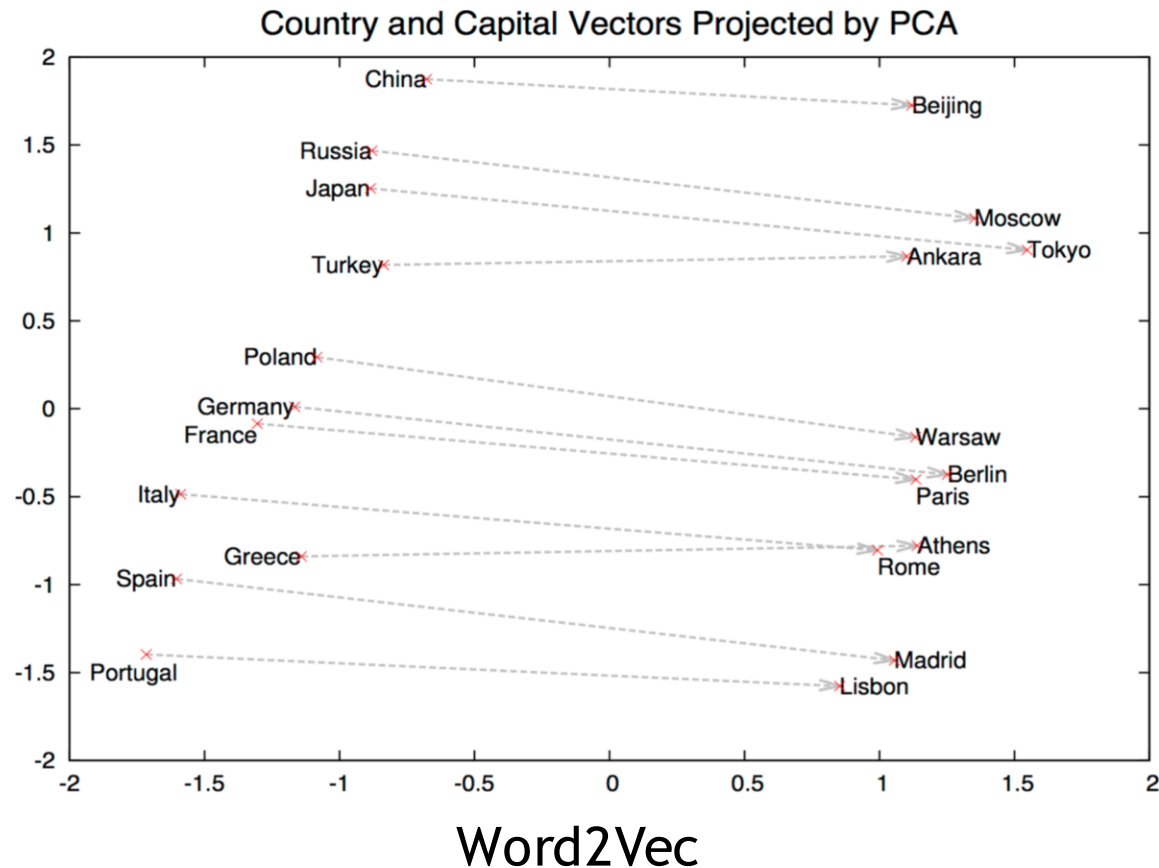
semantic: $v(\text{king}) - v(\text{man}) + v(\text{woman}) \approx v(\text{queen})$

syntactic: $v(\text{kings}) - v(\text{king}) + v(\text{queen}) \approx v(\text{queens})$



Analyse des embeddings

De nombreuses relations sémantiques et syntaxiques entre les mots sont (presque) linéaires dans l'espace d'encastrement !



Similitudes entre les langues

La recette pour construire de grands dictionnaires à partir de petits dictionnaires

Ingrédients:

corpus dans une langue (par exemple, l'anglais)

corpus dans une autre langue (par exemple, l'espagnol)

très petit dictionnaire

cat ↔ gato

cow ↔ vaca

dog ↔ perro

fox ↔ zorro

...

Similitudes entre les langues

La recette pour construire de grands dictionnaires à partir de petits dictionnaires

Ingrédients:

corpus dans une langue (par exemple, l'anglais)

corpus dans une autre langue (par exemple, l'espagnol)

très petit dictionnaire

cat ↔ gato

cow ↔ vaca

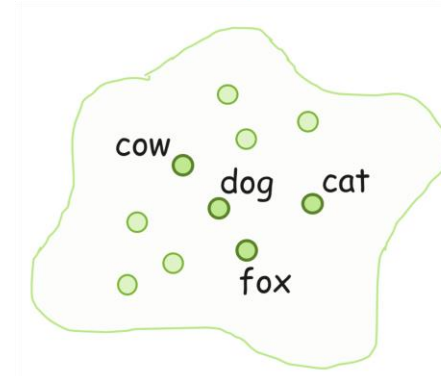
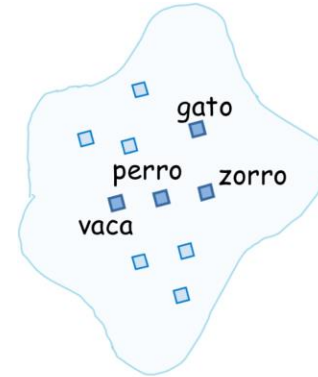
dog ↔ perro

fox ↔ zorro

...

Etape 1:

- Entraîner embeddings pour chaque langage



Similitudes entre les langues

La recette pour construire de grands dictionnaires à partir de petits dictionnaires

Ingrédients:

corpus dans une langue (par exemple, l'anglais)

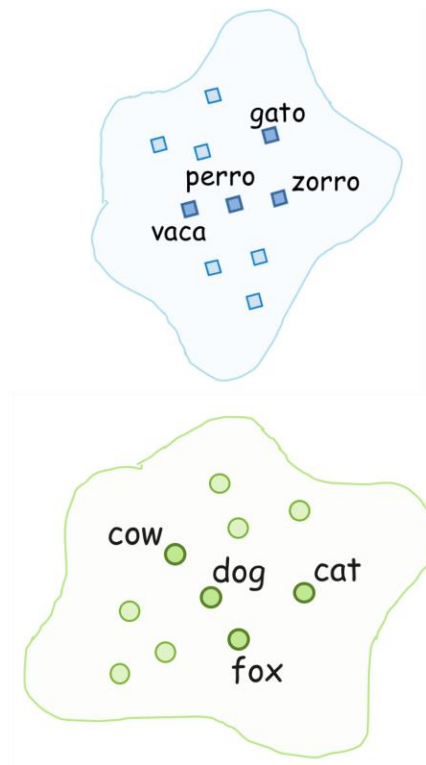
corpus dans une autre langue (par exemple, l'espagnol)

très petit dictionnaire

cat ↔ gato
cow ↔ vaca
dog ↔ perro
fox ↔ zorro
...

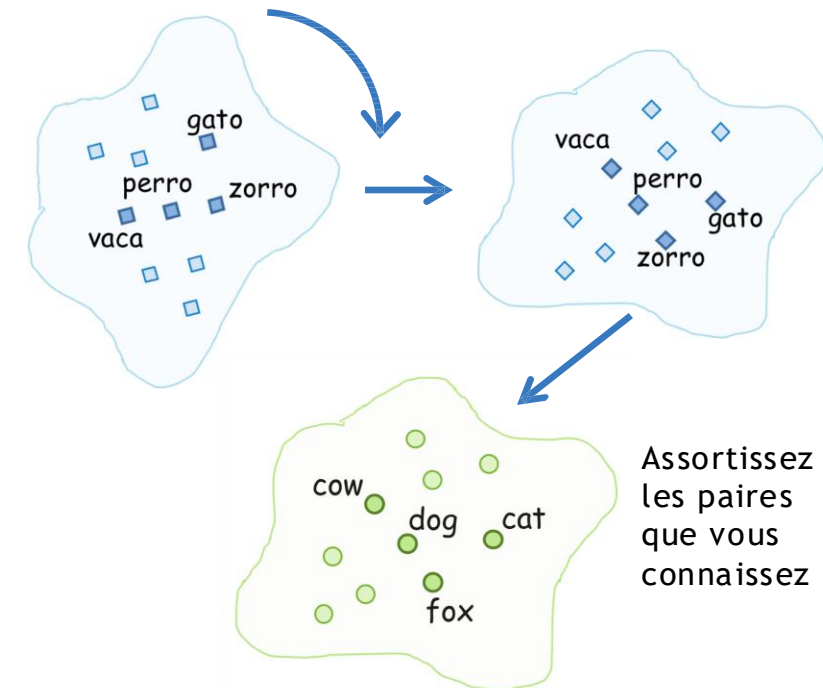
Etape 1:

- Entraîner embeddings pour chaque langue



Etape 2:

- Mappez linéairement l'un des embeddings à l'autre pour correspondre aux mots du dictionnaire



Assortissez les paires que vous connaissez

Similitudes entre les langues

La recette pour construire de grands dictionnaires à partir de petits dictionnaires

Ingrédients:

corpus dans une langue (par exemple, l'anglais)

corpus dans une autre langue (par exemple, l'espagnol)

très petit dictionnaire

cat ↔ gato

cow ↔ vaca

dog ↔ perro

fox ↔ zorro

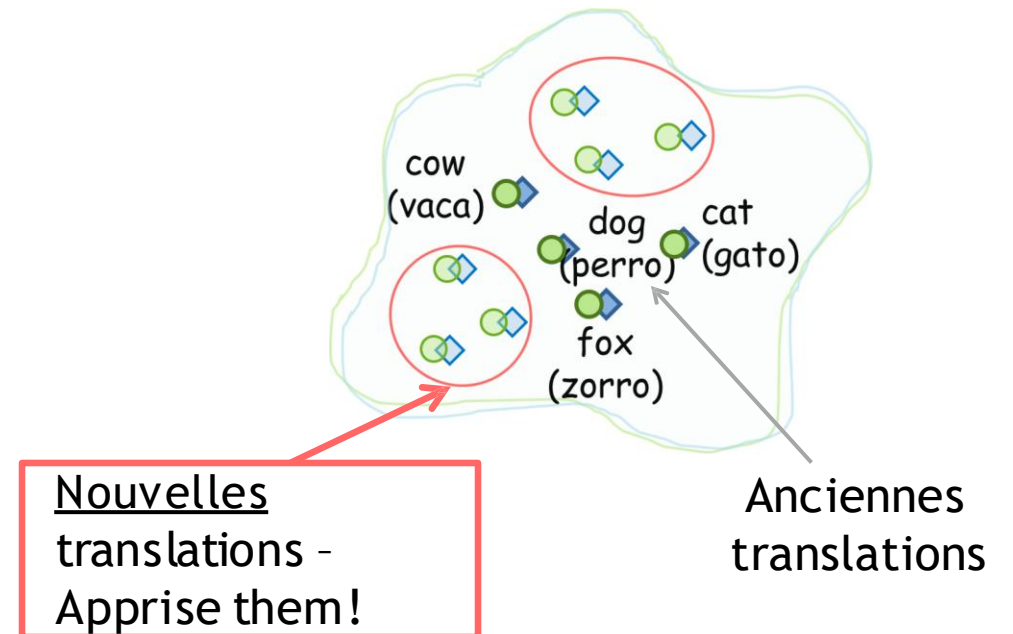
...

Steps1-2:

- Matcher les mots du vocabulaire

Step 3:

- On obtient de nouvelles paires de traduction



A retenir

- On représente le langage avec des tokens
- L'algorithme BPE permet de créer un vocabulaire adapté en sous mots de manière efficace. Utilisé industriellement
- Chaque token est encodé en embedding pour ajouter du sens avant traitement
- Les embeddings sont directement appris à partir des données

Prochain cours

- Découvrir le mécanisme d'**attention** et son importance en NLP



Merci !

Mohamed Abbas KONATE



mohamed-abbas.konate@michelin.com

