



Introduction

Clément Romic (Hugging Face & Inria)

clement.romac@inria.fr

<https://clementromac.github.io/teaching/>

Pourquoi moi ?

- 3e année de doctorat Inria / Hugging Face
 - *Towards autonomous LLM agents with curiosity-driven RL*
 - => beaucoup de (petits) LLM finetunés
 - => beaucoup de RL appliqué au texte
 - => beaucoup d'ingénierie...

Pourquoi moi ?

- 3e année de doctorat Inria / Hugging Face
 - *Towards autonomous LLM agents with curiosity-driven RL*
 - => beaucoup de (petits) LLM finetunés
 - => beaucoup de RL appliqué au texte
 - => beaucoup d'ingénierie...
- Avant ?
 - Formation initiale en informatique (puis Maths/Info)
 - Data Scientist/ML engineer quelques temps (industrie)
 - Ingénieur de recherche quelques temps (académie)

Pourquoi moi ?

- 3e année de doctorat Inria / Hugging Face
 - *Towards autonomous LLM agents with curiosity-driven RL*
 - => beaucoup de (petits) LLM finetunés
 - => beaucoup de RL appliqué au texte
 - => beaucoup d'ingénierie...
- Avant ?
 - Formation initiale en informatique (puis Maths/Info)
 - Data Scientist/ML engineer quelques temps (industrie)
 - Ingénieur de recherche quelques temps (académie)
- Ce que je ne suis pas:
 - un expert en prompting
 - un utilisateur régulier de LLMs
 - un expert dans tous les détails des LLMs de 2024

Pourquoi moi ?

Objectif: vous transmettre des informations qui vont durer, dans un domaine où tout change très (très) rapidement

- 3e année de doctorat Inria / Hugging Face
 - *Towards autonomous LLM agents with curiosity-driven RL*
 - => beaucoup de (petits) LLM finetunés
 - => beaucoup de RL appliqué au texte
 - => beaucoup d'ingénierie...
- Avant ?
 - Formation initiale en informatique (puis Maths/Info)
 - Data Scientist/ML engineer quelques temps (industrie)
 - Ingénieur de recherche quelques temps (académie)
- Ce que je ne suis pas:
 - un expert en prompting
 - un utilisateur régulier de LLMs
 - un expert dans tous les détails des LLMs de 2024

Pourquoi vous ?



Pourquoi ce cours ?

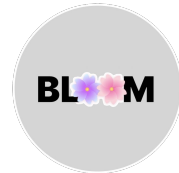
A cause de ChatGPT bien sûr !



A cause de ChatGPT bien sûr !

Plus généralement:

- Nous sommes passés d'outils réservés aux experts à des **outils accessibles à tout le monde**
- Notamment grâce à l'utilisation du langage !

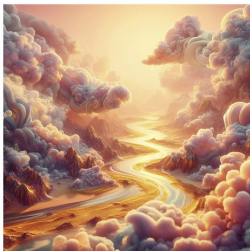


A cause de ChatGPT bien sûr !

Plus généralement:

- Nous sommes passés d'outils réservés aux experts à des **outils accessibles à tout le monde**
- Notamment grâce à l'utilisation du langage !
- A ouvert la porte à plus que des chatbots avec l'idée de **prompt**

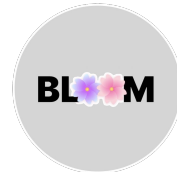
DALL-E 3



MIDJOURNEY 5.2



STABLE XL



Parce que ça a changé le domaine

- L'arrivée des **Transformers (2017)** a fait aux benchmarks de NLP ce que les CNNs (2012) avaient fait à ImageNet...
- Il y a désormais des Transformers (presque) partout en NLP

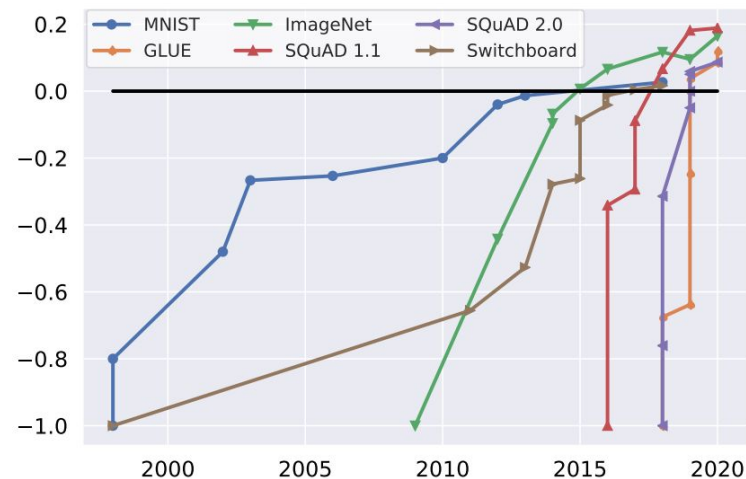


Figure 1: Benchmark saturation over time for popular benchmarks, normalized with initial performance at minus one and human performance at zero.

(Kiel et al., 2021)

Parce que ça a changé le domaine

- L'arrivée des **Transformers (2017)** a fait aux benchmarks de NLP ce que les CNNs (2012) avaient fait à ImageNet...
- Il y a désormais des Transformers (presque) partout en NLP



Figure 1: Benchmark saturation over time for popular benchmarks, normalized with initial performance at minus one and human performance at zero.

(Kiel et al., 2021)

Parce que vous risquez de vous en servir...

Ashish Vaswani

Startup
Verified email at fastmail.com
Deep Learning

TITLE	CITED BY	YEAR
Attention is all you need A Vaswani Advances in Neural Information Processing Systems	141261	2017
Relational inductive biases, deep learning, and graph networks PW Battaglia, JB Hamrick, V Bapst, A Sanchez-Gonzalez, V Zambaldi, ... arXiv preprint arXiv:1806.01261	3802	2018
Self-attention with relative position representations P Shaw, J Uszkoreit, A Vaswani arXiv preprint arXiv:1803.02155	2603	2018
Image transformer N Parmar, A Vaswani, J Uszkoreit, L Kaiser, N Shazeer, A Ku, D Tran International conference on machine learning, 4055-4064	1986	2018
Advances in neural information processing systems A Vaswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, AN Gomez, ... Attention is all you need	1906	2017
Attention Is All You Need. (Nips), 2017 A Vaswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, AN Gomez, ... arXiv preprint arXiv:1706.03762 10, S0140525X16001837	1445	2017
Attention augmented convolutional networks I Balla, P Zeng, A Vaswani, J Shlens, CVL	1341	2019

Cited by

	All	Since 2019
Citations	164572	161484
h-index	46	45
i10-index	64	58

Year	Citations
2018	~1,000
2019	~2,000
2020	~4,000
2021	~8,000
2022	~16,000
2023	~36,000
2024	~32,000

Public access

VIEW ALL

0 articles not available 3 articles available

Based on funding mandates

Déroulé du cours

Déroulé du cours

- **Jour 1:** Des RNNs aux Transformers
 - Tokenization & Embeddings (NLP)
 - Rappels RNNs
 - Attention mechanism
 - Self-Attention and Transformers
- **Jour 2:** LLMs
 - Quizz/Rappels Transformers
 - Encoder-only (e.g. BERT)
 - Decoder-only (e.g. GPT)
 - Prompting
 - Chat models
- TP colab / Jupyter tout le long
 - [Jour 1](#)
 - [Jour 2](#)

Déroulé du cours

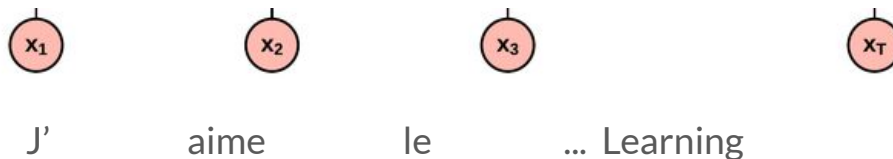
- Objectif du cours: vous donner les clés pour comprendre comment ça marche, à vous d'aller plus loin si vous voulez
- Evaluation:
 - Quelques questions au QCM
- Feedback apprécié !
 - Formulaire anonyme à la fin du cours
- Tout est sur mon site web: <https://clementromac.github.io/teaching/>

Un peu de Natural Language Processing

Tokenization

Objectif global: utiliser du texte comme entrée

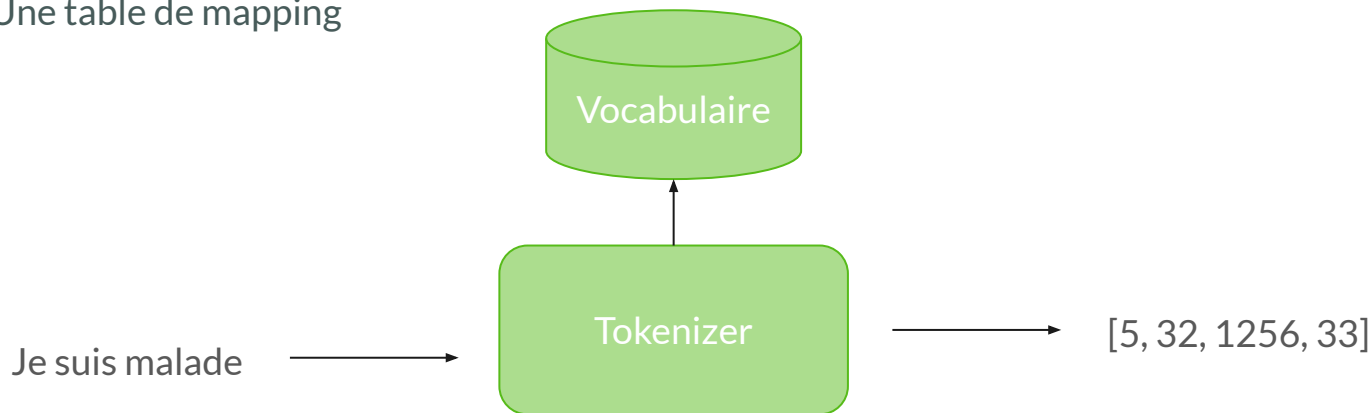
Etape 1: Passer d'une séquence de mots à une séquence de symboles connus



Tokenization

Tokenizer:

- Un “vocabulaire” de symboles
- Une table de mapping



Tokenization

Word-level mapping:

Je suis malade \longrightarrow ["Je": 5, "suis": 32, "malade": 1256]

=> Vocabulaire relativement petit, aucun partage de racine

Tokenization

Word-level mapping:

Je suis malade \longrightarrow ["Je": 5, "suis": 32, "malade": 1256]

=> Vocabulaire relativement petit, aucun partage de racine

Character-level mapping:

Je suis malade \longrightarrow ["J": 10, "e": 5, "s": 18, ...]

=> Vocabulaire très petit mais peu informatif

Tokenization

Word-level mapping:

Je suis malade → ["Je": 5, "suis": 32, "malade": 1256]

=> Vocabulaire relativement petit, aucun partage de racine

Character-level mapping:

Je suis malade → ["J": 10, "e": 5, "s": 18, ...]

=> Vocabulaire très petit mais peu informatif

Tokenizers aujourd'hui utilisés:

- WordPiece (*Schuster et al., 2012*)
 - BERT
- Byte Pair Encoding (*Sennrich et al., 2018*)
 - GPT
- SentencePiece (*Kudo et al., 2018*)
 - T5, Llama

Note: GPT-2: 50k tokens

Tokenization

Tokens spéciaux:

Je suis malade \longrightarrow [“</s>”: 34, “Je”: 5, “suis”: 32, “malade”: 1256, “<s>”]

- <pad> => pad (+ mask) pour avoir des batchs de même taille
- </s> => début de séquence
- <s> => fin de séquence
- <unk> => symbole inconnu (hors de la table)
- ...

=> Dépend du tokenizer

Tokenization



TP: Partie 1

Tokens spéciaux:

Je suis malade \longrightarrow [“</s>”: 34, “Je”: 5, “suis”: 32, “malade”: 1256, “<s>”]

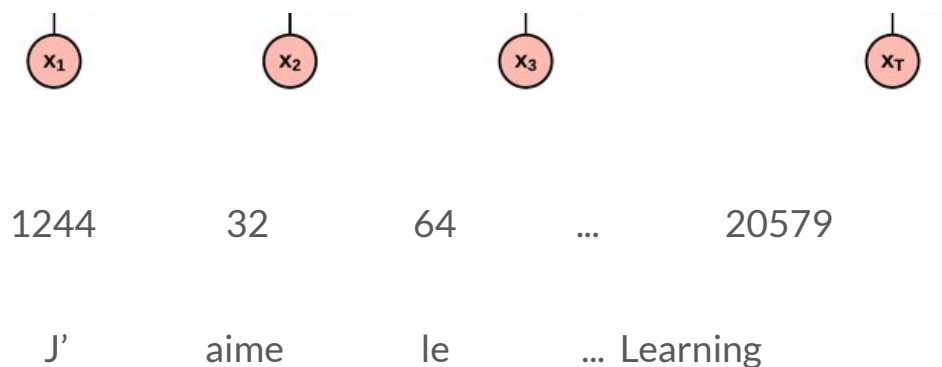
- <pad> => pad (+ mask) pour avoir des batchs de même taille
- </s> => début de séquence
- <s> => fin de séquence
- <unk> => symbole inconnu (hors de la table)
- ...

=> Dépend du tokenizer

Word Embeddings

Objectif global: utiliser du texte comme entrée

Etape 2: Passer d'une séquence de tokens à une séquence de vecteurs



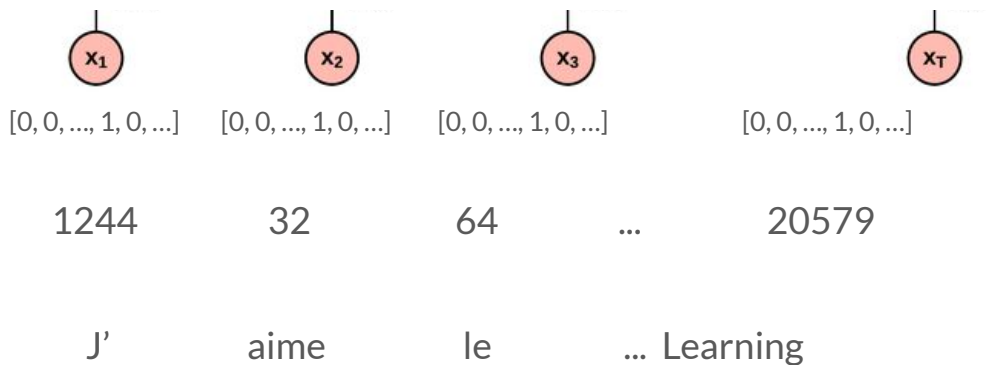
Word Embeddings

Objectif global: utiliser du texte comme entrée

=> Vecteurs sparses
=> Aucun intérêt sémantique

Etape 2: Passer d'une séquence de tokens à une séquence de vecteurs

Solution naïve:
OneHotEncoding

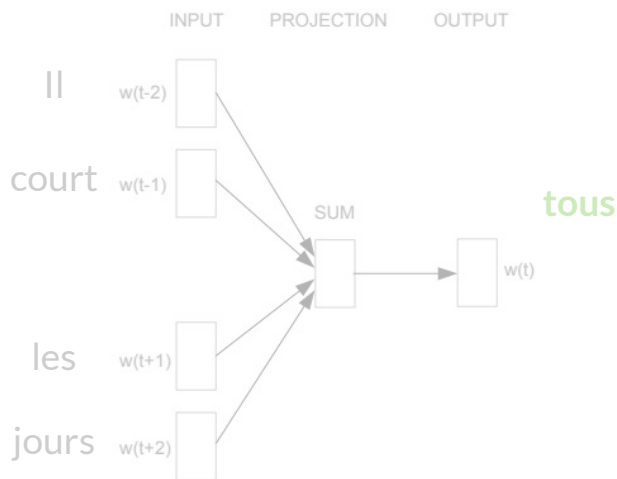


Word2Vec *(Mikolov et al., 2013)*

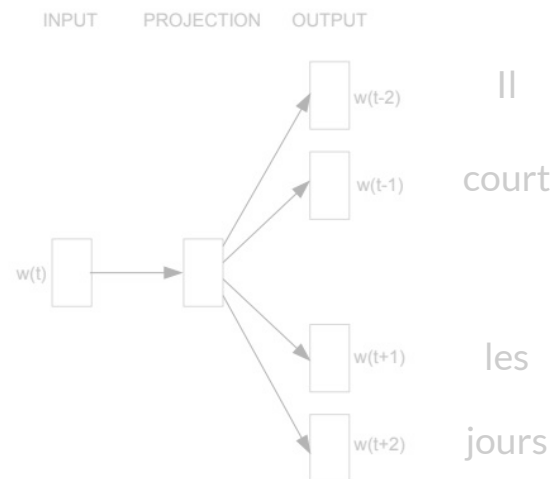
Intuition: Apprendre à représenter un mot à partir de son contexte

1) On utilise un vecteur **OneHotEncoding** pour chaque mot

2) Deux approches:



CBOW



Skip-gram

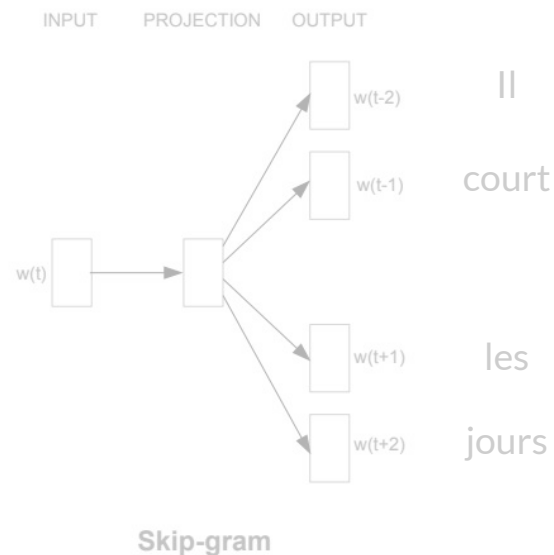
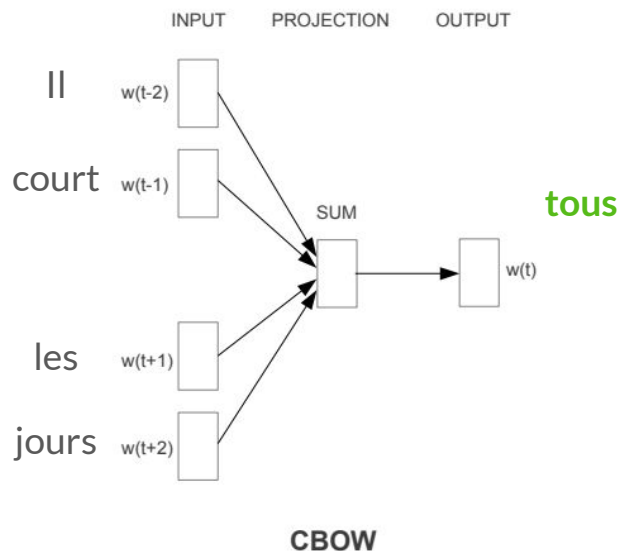
Word2Vec *(Mikolov et al., 2013)*

Intuition: Apprendre à représenter un mot à partir de son contexte

1) On utilise un vecteur **OneHotEncoding** pour chaque mot

2) Deux approches:

CBOW: On passe chaque mot du contexte dans une **couche linéaire partagée**, on fait la **moyenne de tous les vecteurs** et on **prédit le mot attendu**



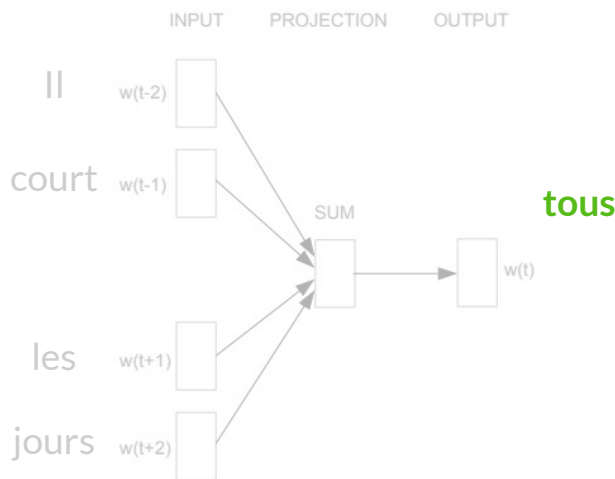
Word2Vec *(Mikolov et al., 2013)*

Intuition: Apprendre à représenter un mot à partir de son contexte

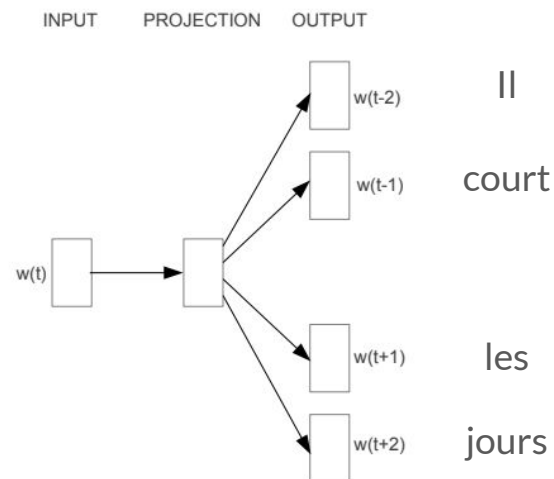
1) On utilise un vecteur **OneHotEncoding** pour chaque mot

2) Deux approches:

Skip-gram: On passe le mot principal dans une couche linéaire, on essaie de prédire chacun des mots du contexte



CBOW

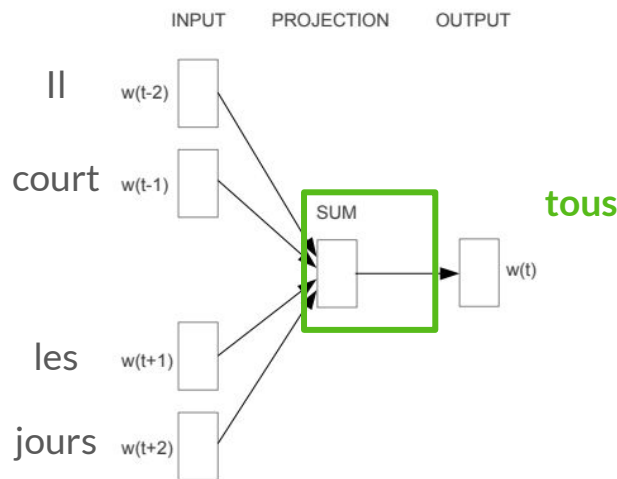


Skip-gram

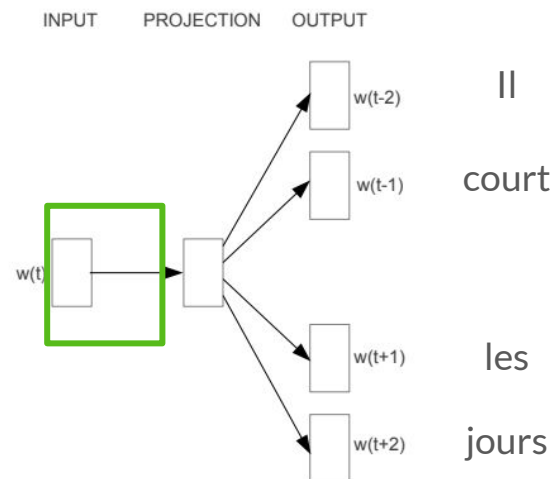
Word2Vec *(Mikolov et al., 2013)*

Intuition: Apprendre à représenter un mot à partir de son contexte

- 1) On utilise un vecteur **OneHotEncoding** pour chaque mot
- 2) Deux approches
- 3) On retient le vecteur obtenu avec le mot principal associé => **lookup table**



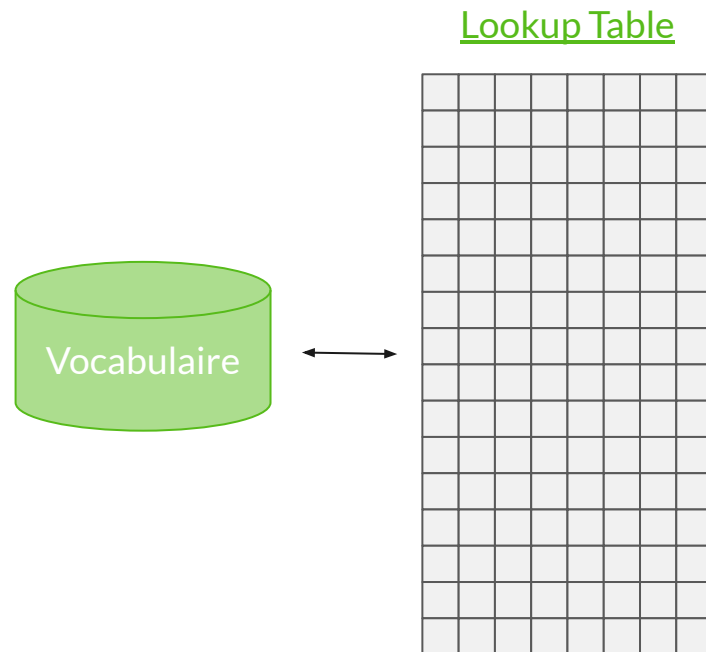
CBOW



Skip-gram

Embedding lookup

- On a donc une **table associant chaque token à son embedding**
- On peut **utiliser ces embedding** en entrée pour entraîner notre modèle



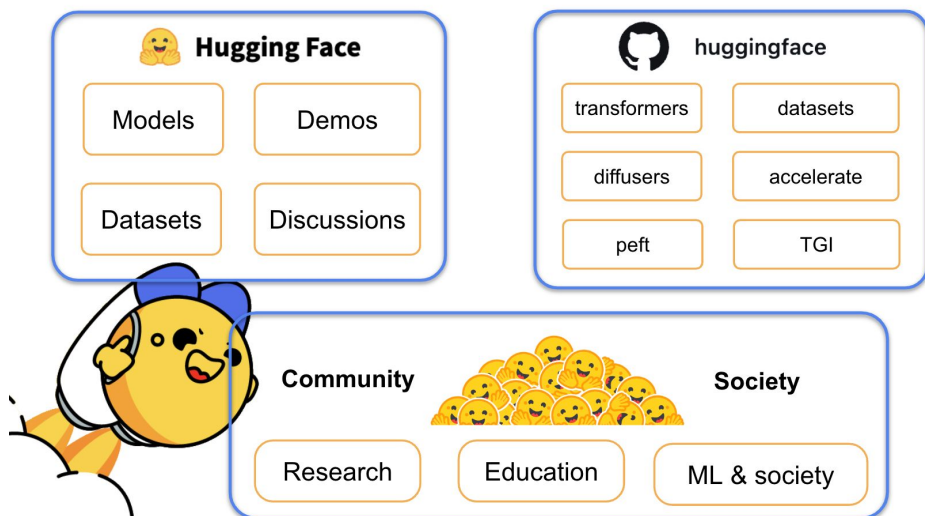
Word2Vec + Language MModel

Dans ce cours (et généralement):

On initialise la table aléatoirement et les
embeddings sont appris en même temps que le
modèle

Outils open-source

Outils Open-Source *(non exhaustif)*



Outils Open-Source *(non exhaustif)*

