Introduction

Clément Romac (Hugging Face & Inria)

clement.romac@inria.fr
https://github.com/ClementRomac/Teaching/tree/main/ENSC3A_LLMs_2024-2025

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 ?

- 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

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

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**

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.

(Kiela et al., 2021)

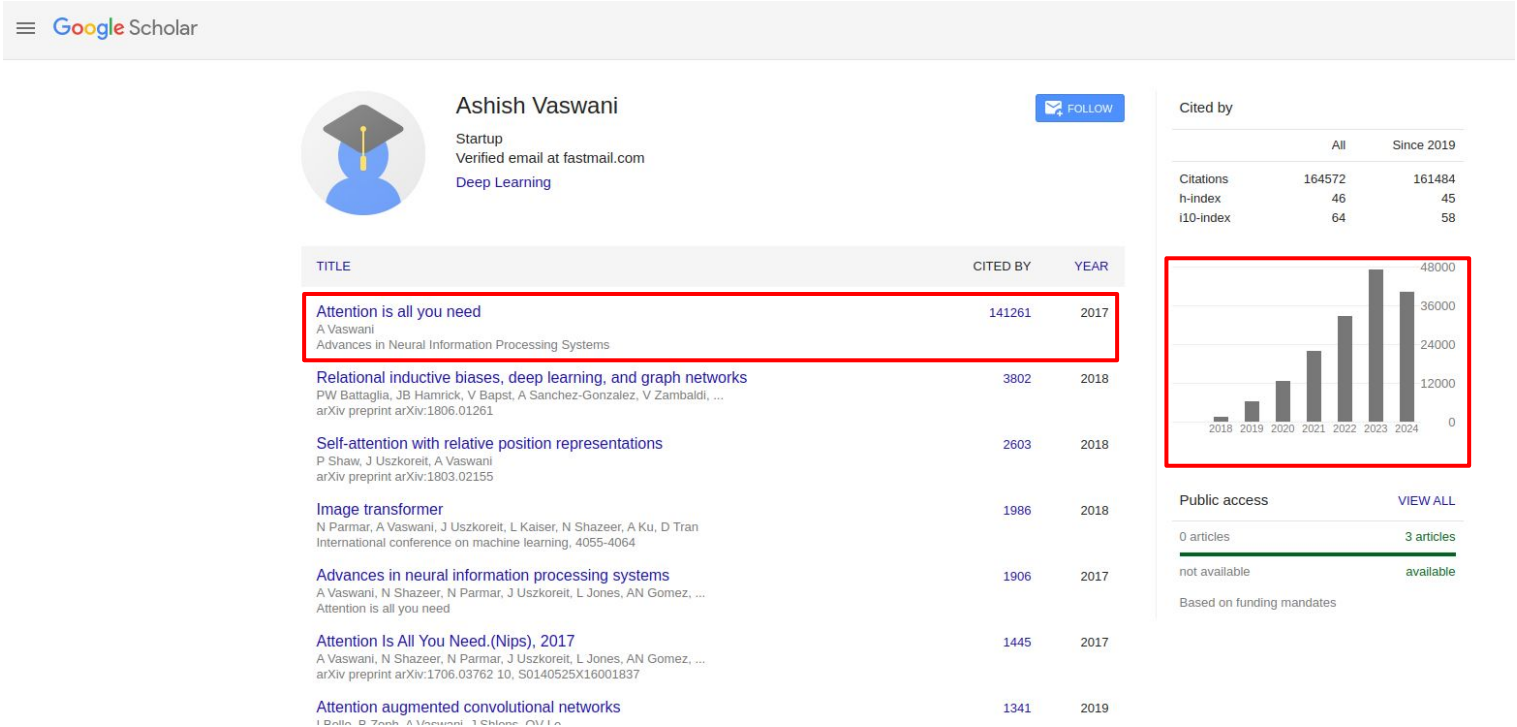
Parce que ça a changé le domaine

- 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.

(Kiela et al., 2021)

Parce que vous risquez de vous en servir…



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

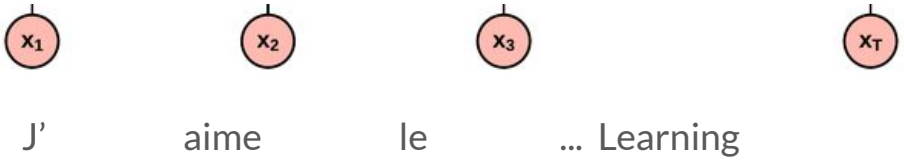
- Tout est sur mon site web: https://clementromac.github.io/teaching/

Un peu de Natural Language Processing

Tokenization

Objectif global: utiliser du texte comme entrée

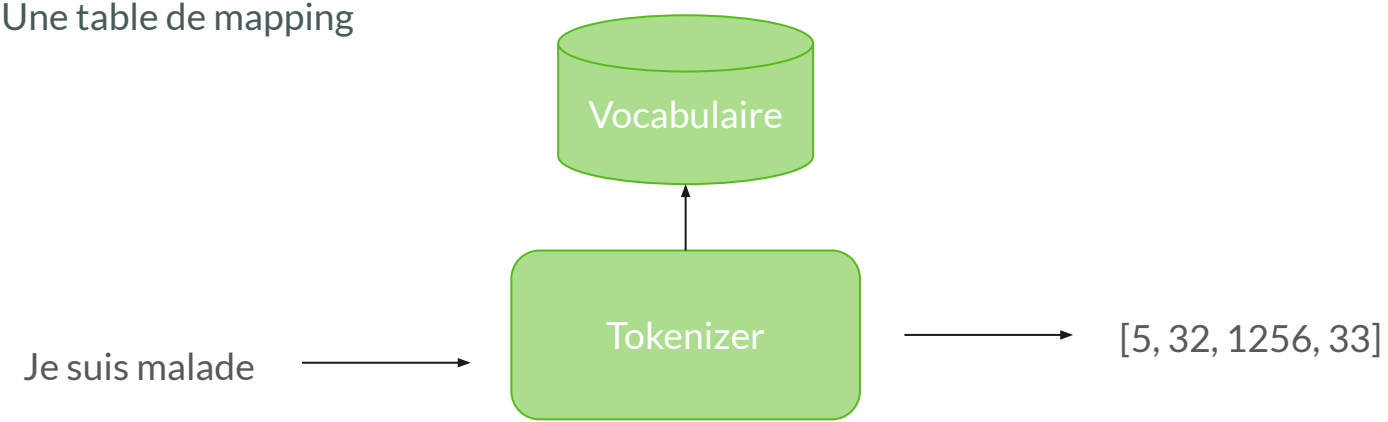
Etape 1: Passer d'une séquence de mots à une séquence de symboles connus



Tokenization

Tokenizer:

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



Tokenization

Word-level mapping:

Je suis malade ⟶ [“Je”: 5, “suis”: 32, “malade”: 1256]

=> Vocabulaire relativement petit, aucun partage de racine

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

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

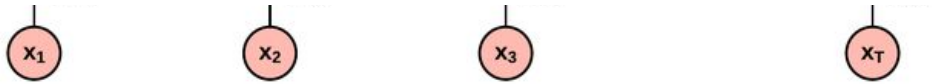
- <pad> => pad (+ mask) pour avoir des batchs de même taille
- </s> => début de séquence
- <s> => fin de séquence
- <unk> => symbole inconnu (hors de la table)
- …

=> 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



1244 32 64 ... 20579

J' aime le ... Learning

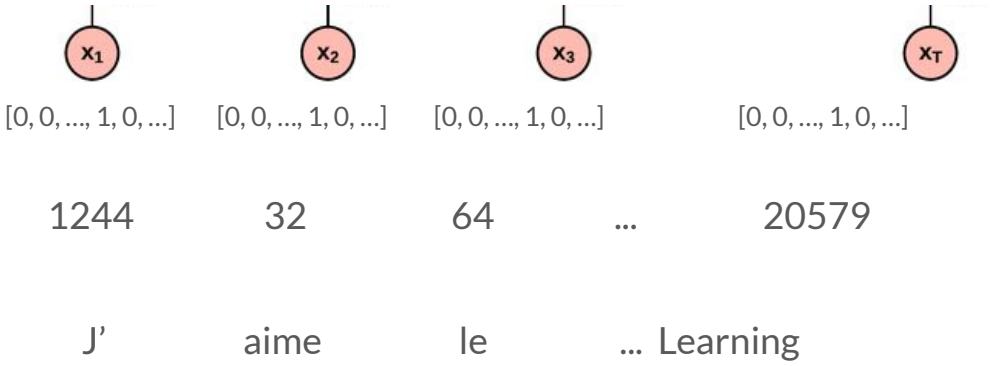
Word Embeddings

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

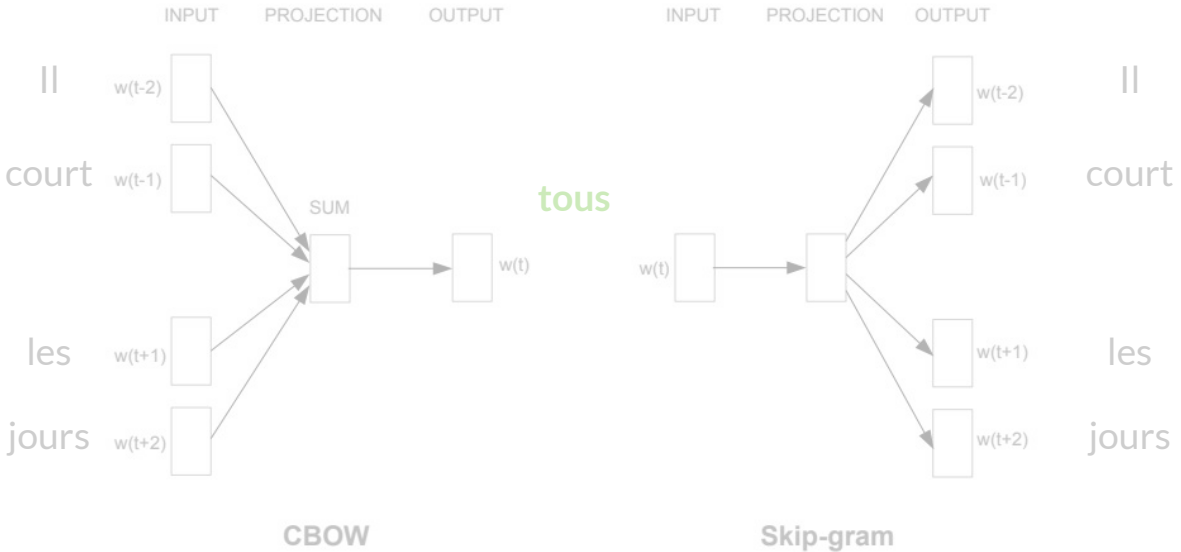


Word2Vec *(Mikolov et al., 2013)*

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

1) On utilise un vecteur OneHotEncoding pour chaque mot

2) Deux approches:



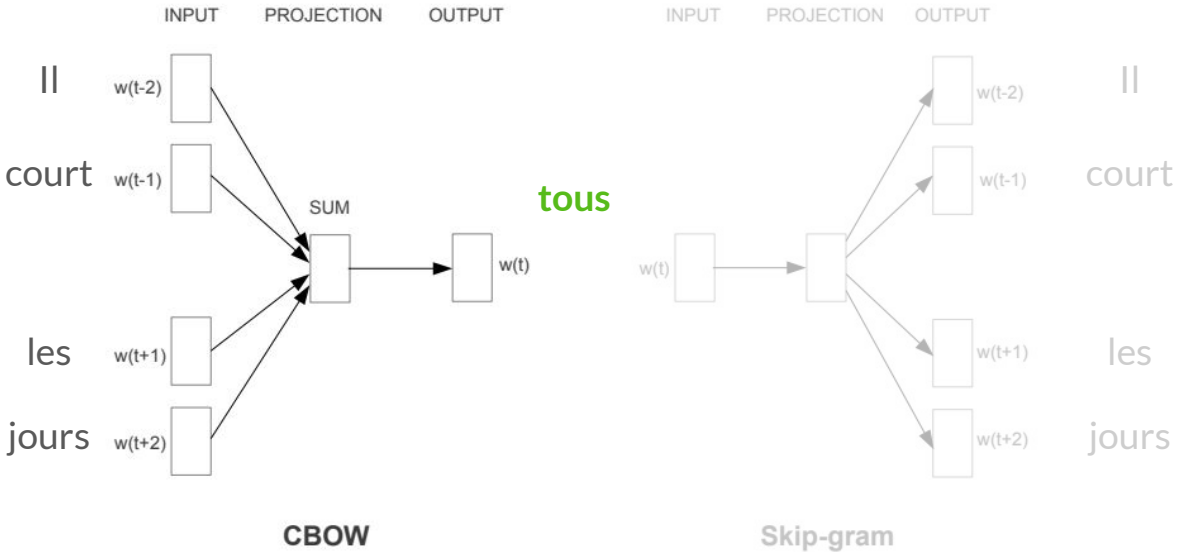
Word2Vec *(Mikolov et al., 2013)*

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



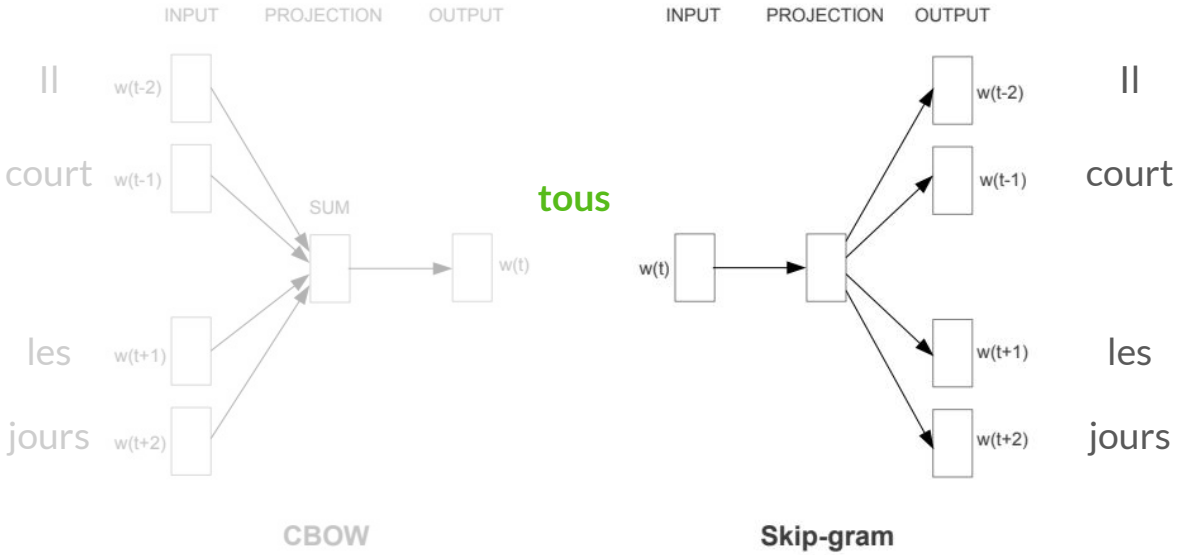
Word2Vec *(Mikolov et al., 2013)*

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



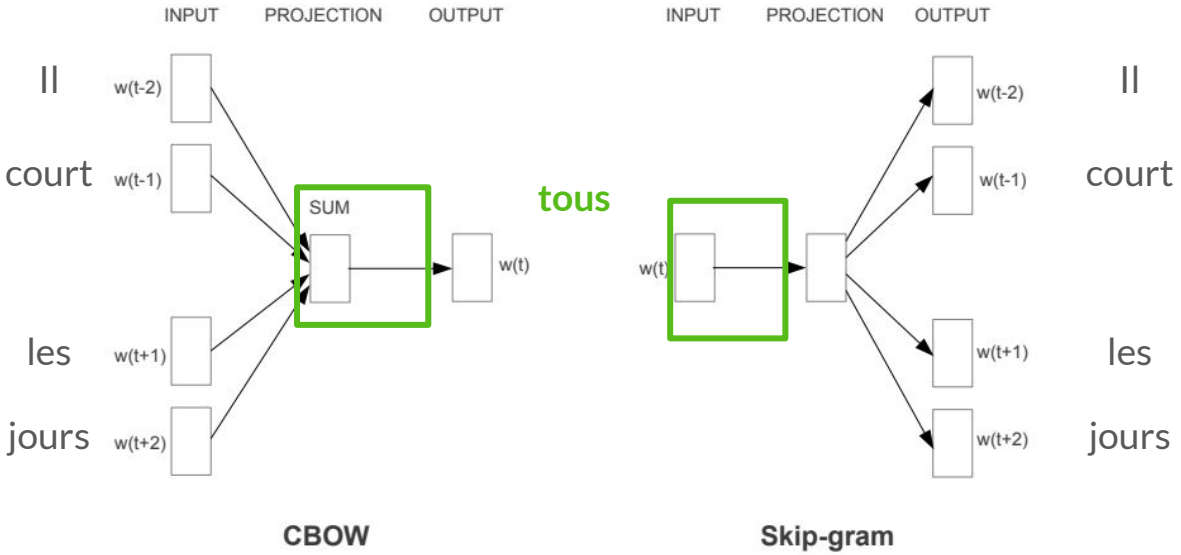
Word2Vec *(Mikolov et al., 2013)*

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

1) On utilise un vecteur OneHotEncoding pour chaque mot

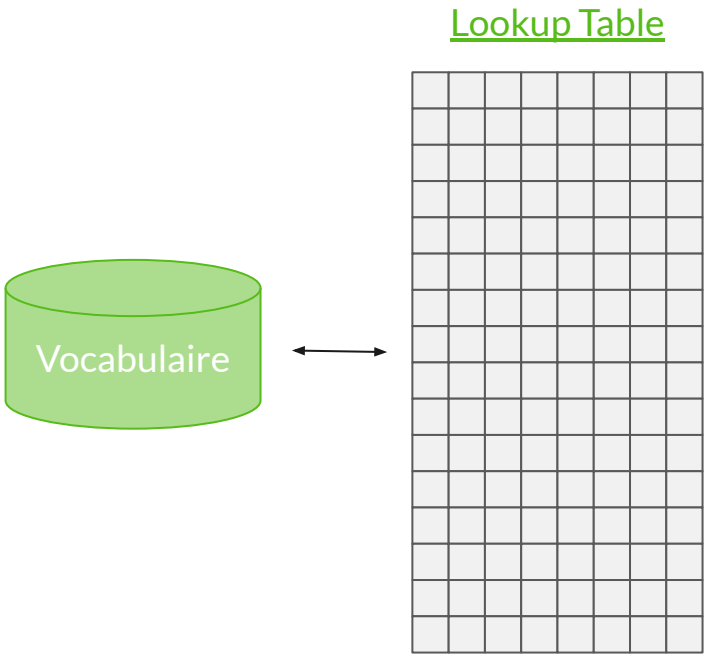
2) Deux approches

3) On retient le vecteur obtenu avec le mot principal associé => **lookup table**



Embedding lookup

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



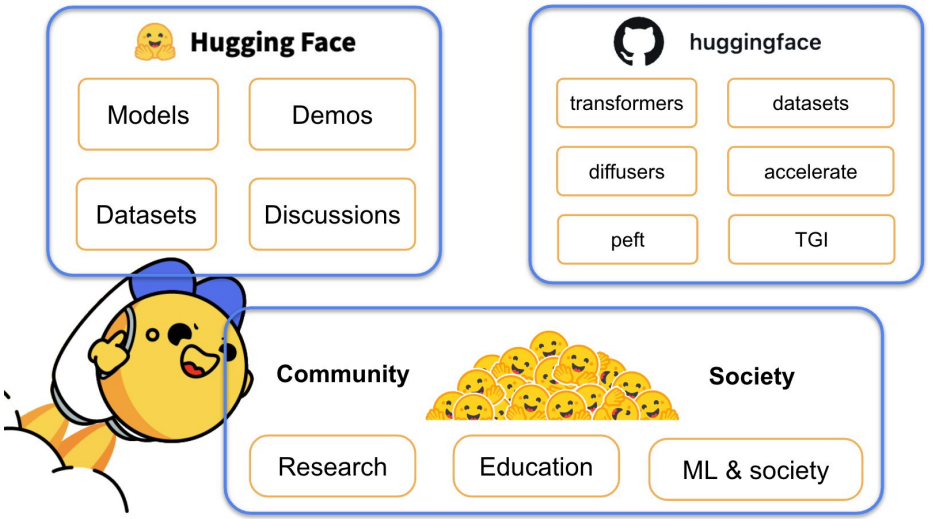
Word2Vec + Language MOdel

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)*

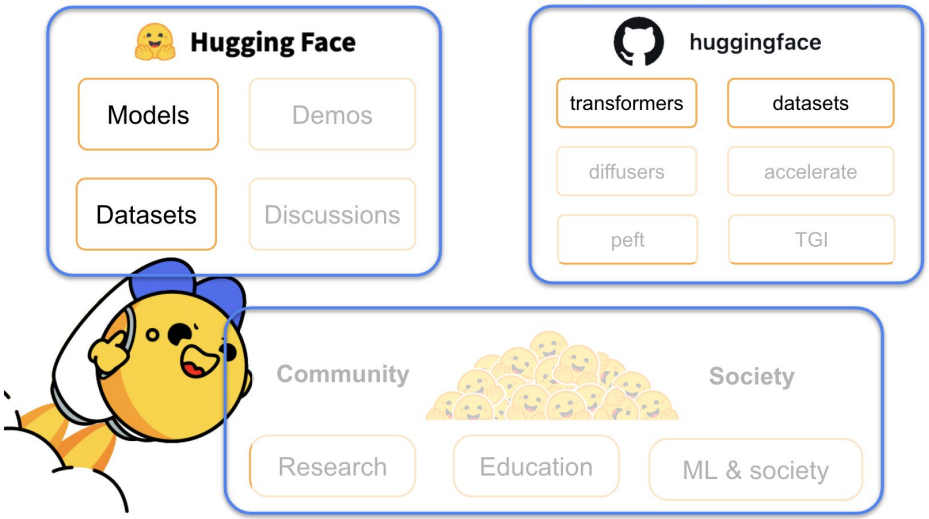


spaCy

tiktoken

SentencePiece

Outils Open-Source *(non exhaustif)*



spaCy

tiktoken

PyTorch

SentencePiece