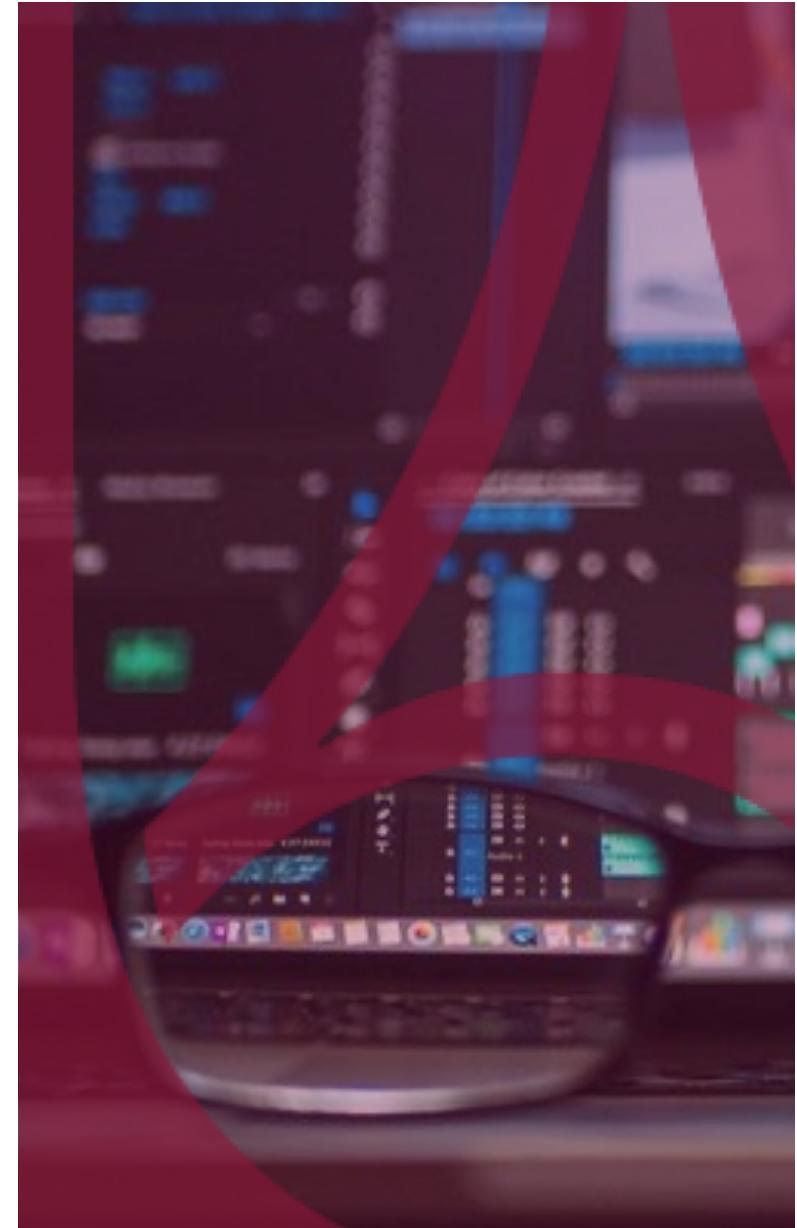


Traitement Automatique de la Langue Naturelle

M2 Data Science • Année 2020 / 2021

Marie CANDITO - Antoine SIMOULIN



Ressources



<https://moodle.u-paris.fr/course/view.php?id=11048>



<https://github.com/AntoineSimoulin/m2-data-sciences>



<m2midsunivdeparis.slack.com>

Icon made by [Freepik](#), [Becris](#), [Smashicons](#), [Pixel perfect](#), [srip](#), [Adib](#), [Flat Icons](#), [Vitaly Gorbachev](#), [Becris](#) from www.flaticon.com

4

Séance #10

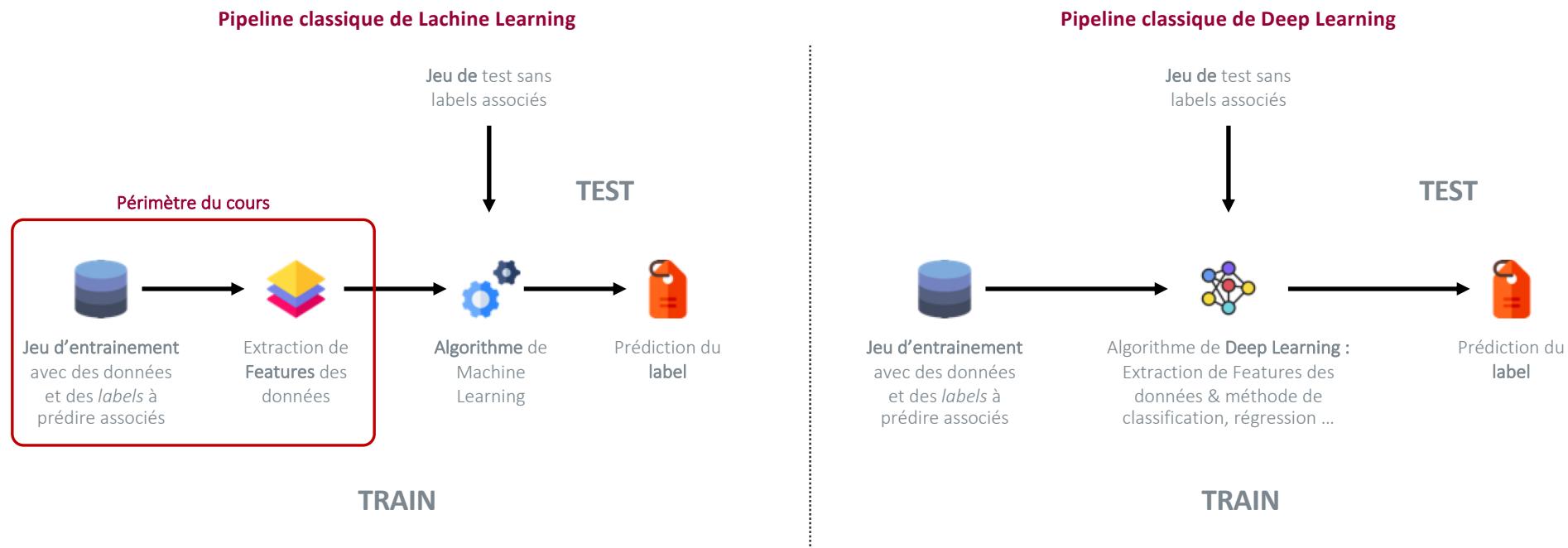


Attention, il s'agit d'un cours d'introduction où l'on énumère simplement des méthodes de Deep Learning pour le NLP. Les supports présentés cherchent simplement à illustrer des notions qui seront approfondies pendant le cours d'Introduction au Deep Learning.

Le Deep Learning



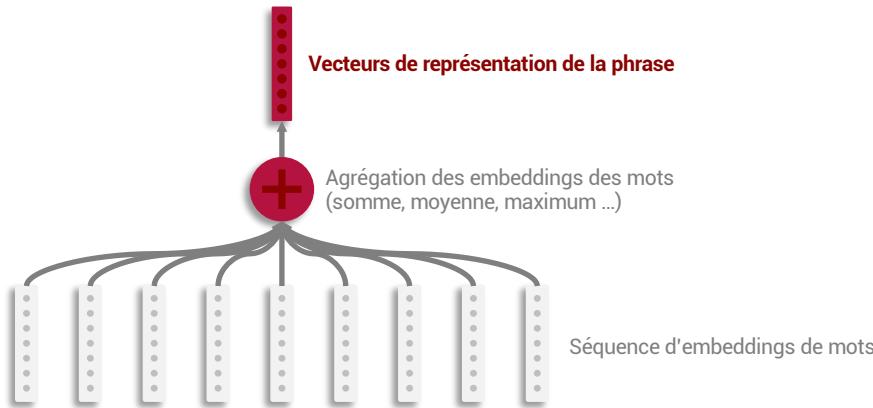
Le **Deep Learning** est une branche du machine learning qui regroupe des algorithmes spécifiques : les **réseaux de neurones**. Les paramètres de ces architectures sont appris par *back-propagation*. Il existe de nombreuses spécificités du Deep Learning. L'aspect qui nous intéresse ici c'est que ce type de méthode présente généralement un pipeline simplifié où les **features et les méthodes font partie de la même architecture et sont appris conjointement**.



Le Deep Learning pour le NLP



Pendant le cours et les TPs, nous avons vu des méthodes pour transformer une séquence d'embeddings en un vecteur de représentation. Nous avons utilisé des méthodes d'agrégations comme la moyenne, la somme ... En pratique, on peut utiliser des méthodes d'agrégations plus subtiles.

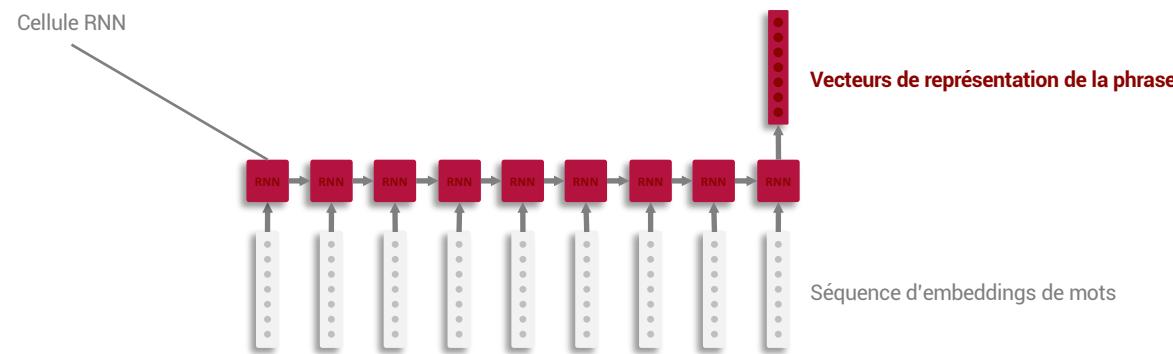


[1] Sanjeev Arora, Yingyu Liang, Tengyu Ma: *A Simple but Tough-to-Beat Baseline for Sentence Embeddings*. ICLR (Poster) 2017



Les réseaux de neurones récurrents (1/4)

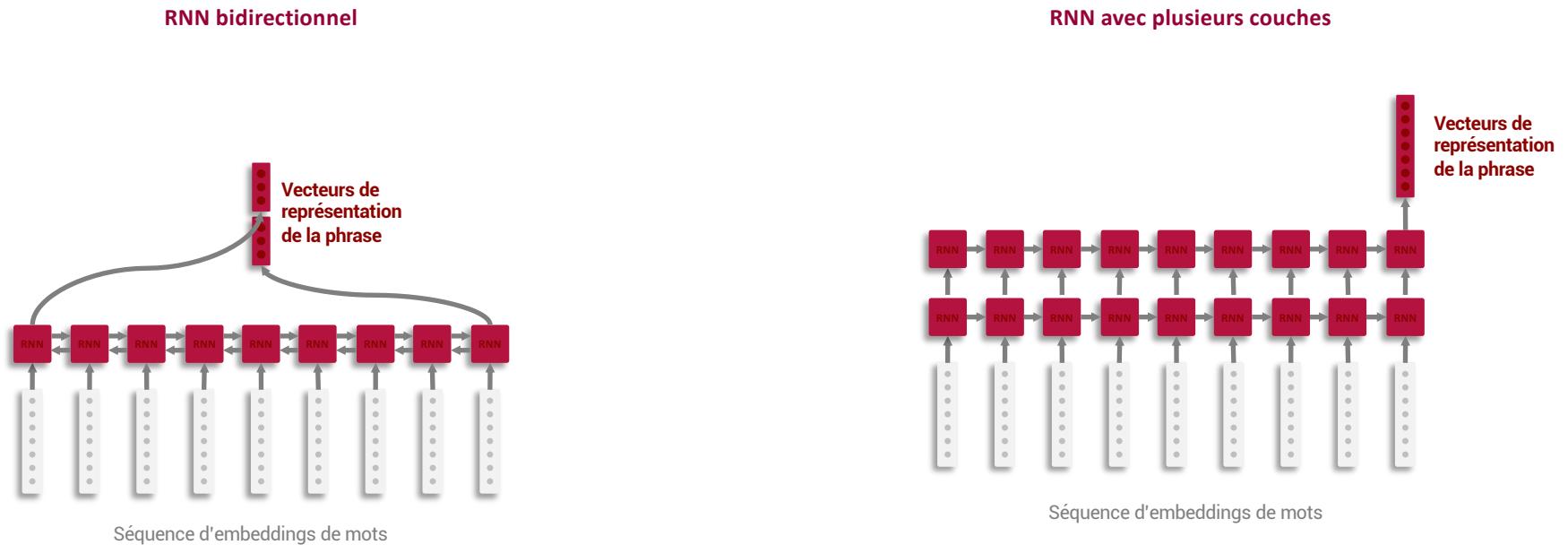
Les réseaux de neurones récurrents modélisent les phrases comme des séquences d'embeddings de mots. Ils traitent l'entrée séquentiellement. A chaque étape, le vecteur de sortie est calculé en fonction de l'embedding du mot courant et de l'état caché précédent.





Les réseaux de neurones récurrents (2/4)

Il existe **plusieurs variantes** des Recurrent Neural Network (RNN). Les cellules **LSTM**^[1] et **GRU**^[2] incluent un **mécanisme de « mémoire »** qui permet de modéliser efficacement les **relations longues distances**. Par ailleurs, il est possible de considérer des RNN **bidirectionnels**. Dans ce cas, le vecteur de représentation correspond à la concaténation de ceux obtenus en parcourant la phrase dans les deux sens. On peut aussi considérer des RNN à plusieurs couches pour lesquels la sortie à chaque étape correspond à l'entrée d'un nouveau RNN.



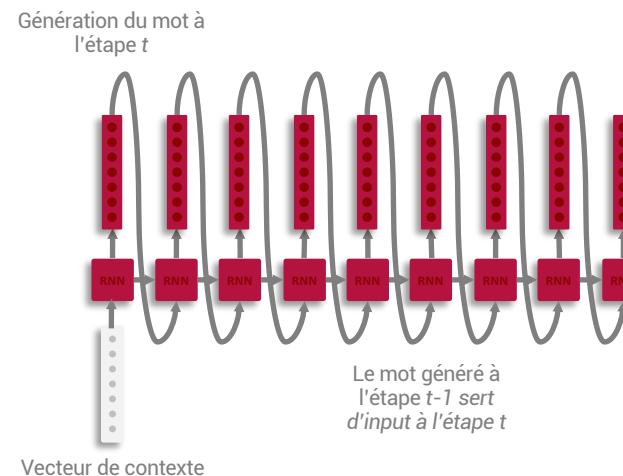
[1] Sepp Hochreiter, Jürgen Schmidhuber: **Long Short-Term Memory**. *Neural Comput.* 9(8): 1735-1780 (1997)

[2] Kyunghyun Cho, Bart van Merriënboer, Çaglar Gülcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio: **Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation**. *EMNLP 2014*: 1724-1734



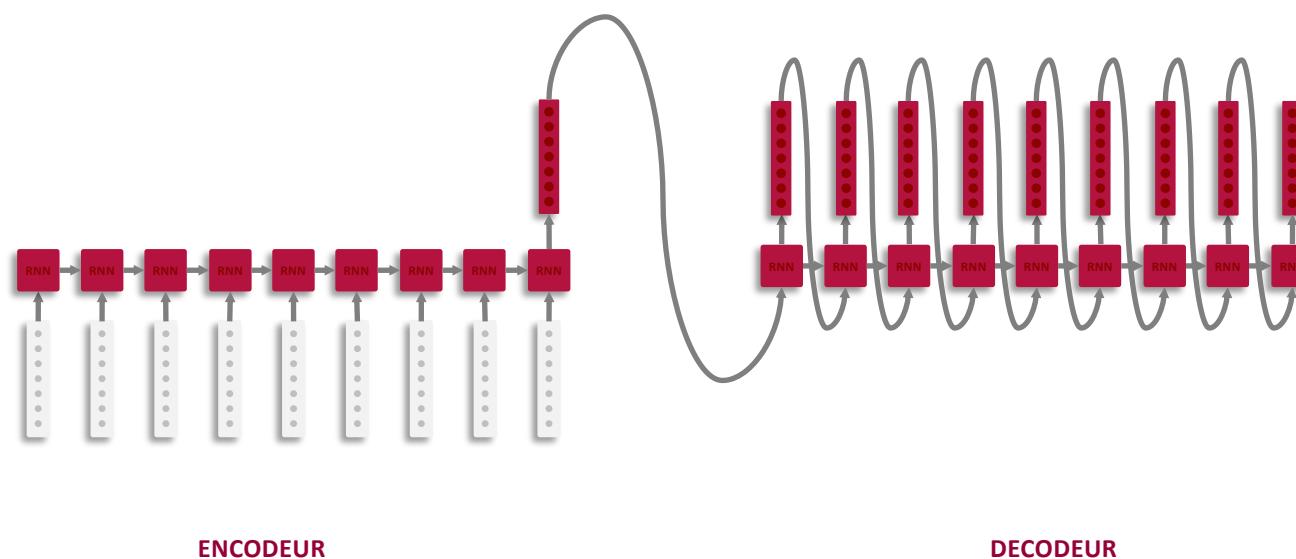
Les réseaux de neurones récurrents (3/4)

Nous avons vu qu'il est possible d'utiliser les RNN pour encoder une séquence. A l'inverse, il est également possible de les utiliser pour générer du texte à partir d'un vecteur. Pour cela le mot généré à chaque étape sert d'input pour la génération du mot suivant.



Les réseaux de neurones récurrents (4/4)

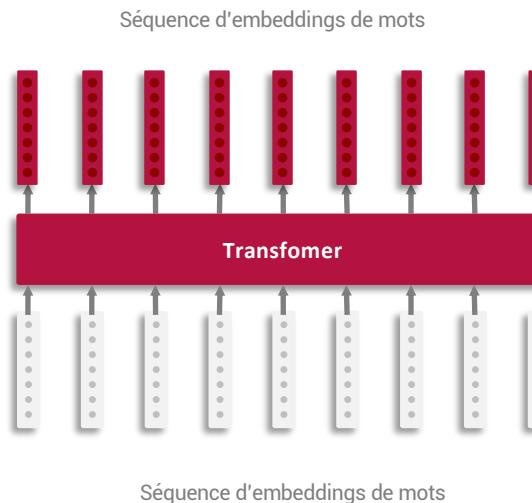
Il est possible de combiner le module encodeur et le module décodeur pour créer une **architecture Sequence-to-Sequence**. Ce type d'architecture sert de base pour de nombreux cas d'usages de NLP. Pour le **résumé automatique**, on encode le texte et on cherche à décoder le résumé. Pour la **traduction automatique**, on encode le texte dans une langue et cherche à le décoder dans une autre. On peut également l'utiliser comme un auto-encodeur ou l'on cherche à encoder un texte puis à le décoder à partir du vecteur de représentation. Ce type d'architecture a été déclinée selon de nombreux raffinements.



Les transformers (1/2)



Les architectures Sequence-to-Sequence ont aussi été déclinées sans l'utilisation de réseaux récurrents. En particulier avec **l'architecture transformers** [1]. Sans rentrer dans les détails, cette dernière s'appuie sur **l'opération d'attention**. Elle est plus facile à paralléliser que les RNN.

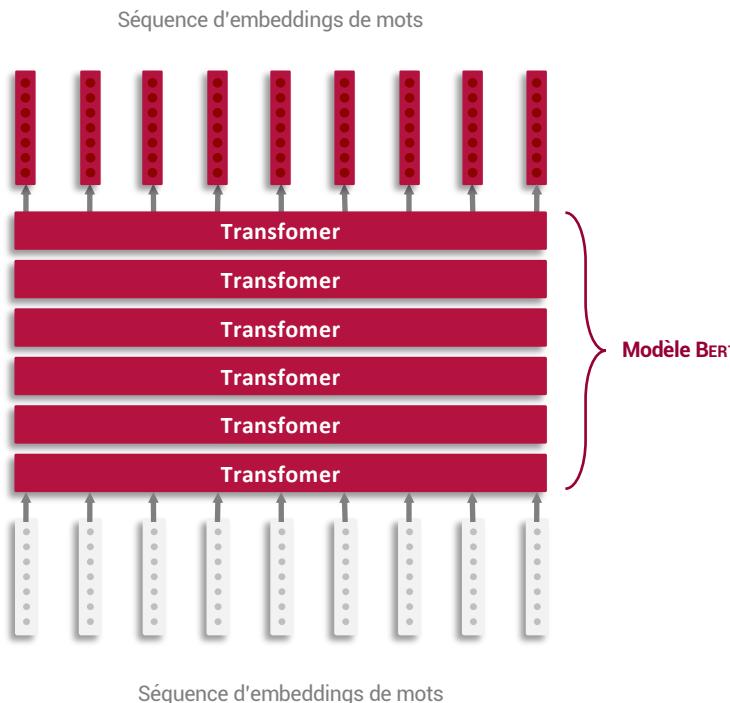


[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin: *Attention is All you Need*. NIPS 2017: 5998-6008



Les transformers (2/2)

Les architectures transformers ont inspiré le modèle **BERT** qui consiste en une **succession de transformers**. Ce modèle présente également la spécificité d'être **pré-entraîné** : les poids sont initialisés en entraînant le modèle sur une tâche auto-supervisée. En l'occurrence prédire des mots masqués, on parle de modèle de langue. Ce modèle est très versatile et **peut être décliné pour quasiment tous les cas d'usages du NLP**.



[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova: **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.** NAACL-HLT (1) 2019: 4171-4186

5

Bilan

Systèmes Symboliques vs Statistiques

Systèmes Symboliques vs Statistiques

Les systèmes statistiques d'appuient sur un **corpus d'exemples** pour apprendre la relation entre les labels recherchés et les représentations du texte.

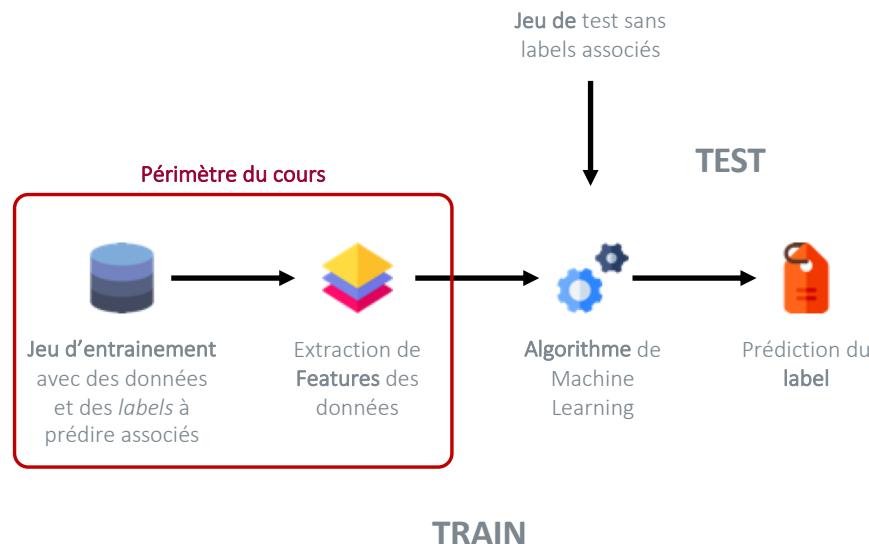
Ils ne supposent pas d'expertise linguistique.

Ils permettent de résoudre une **multitude de cas d'usages** avec un pipeline de traitements unifié.

Ils permettent généralement d'obtenir de meilleures performances mais sont moins intelligibles.

Cours 1 : Un pipeline de traitement unique pour traiter de nombreux cas d'usages

Pipeline classique de Lachine Learning



Applications



Moteurs de recherche



Analyse de sentiments



Systèmes de Questions/
Réponses



Résumé automatique



Traduction automatique



Extraction d'information



Génération de texte



Classification de texte



Exploration de Topics



Vectorisation du texte



Vectorisation du texte

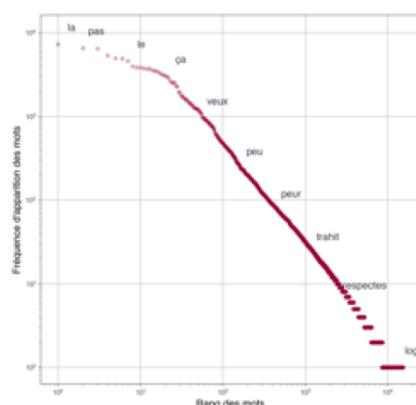
Nous avons vu **deux méthodes de vectorisation** du texte. On représente un document $d \in D$ du corpus par un vecteur e de taille N , la taille du vocabulaire. Chacune des coordonnées i du vecteur correspond à un mot w_i du vocabulaire.

Dans la représentation **Bag-Of-Word (BoW)**, chaque coordonnée du vecteur correspond à tf , la **fréquence du mot dans le document** $f(w_i) = tf(w_i)$.

La distribution des mots empiriques suit la **loi de Zipf**. Si on classe les mots en fonction de leur fréquence d'apparition dans le corpus, cette dernière décroît selon une loi $\propto \frac{1}{N}$.

Pour attribuer moins de poids au mots récurrents mais porteurs de peu d'information, on utilise le TF-IDF. chaque coordonnée du vecteur est données par $f(w_i) = tf(w_i) \times idf(w_i)$ avec $idf(w_i) = \log\left(\frac{N}{df_{w_i}}\right)$ et df_{w_i} le nombre de documents du corpus où apparaît le mot.

Cours 1 : La distribution empirique des mots, la loi de Zipf. **Cours 1 et TP1 :** Les méthodes de vectorisation statistiques : Bag-Of-Word, TF-IDF.



Loi de Zipf pour l'œuvre de Jul

Jean mange une pomme et une orange

Vecteur de fréquence

3	<UNK>
0	dans
0	est
0	jardin
0	le
0	sont
1	pomme
0	pommier
1	mange
2	une

Vecteur de taille (1,N)
N = taille du vocabulaire

Vecteur BoW : On associe à chaque mot, sa fréquence d'apparition

Jean mange une pomme et une orange

0.0	<UNK>
3	dans
0	est
0	jardin
0	le
0	sont
0	pomme
1.2	pommier
0	mange
1.8	une
0.2	

Vecteur de taille (1,N)
N = taille du vocabulaire

Vecteur TF-IDF : On associe à chaque mot, son score TF-IDF

Classification



Classification

Une fois le texte vectorisé, il est possible d'utiliser tous les outils du machine learning pour exploiter les jeux de données.

De nombreux problèmes peuvent être ramenés à des **tâches de classification ou de régression** (détection de Spams dans les emails, analyse de sentiments dans les commentaires utilisateurs ...)

Ce type de méthodes sont très sensibles aux prétraitements et notamment à la **standardisation du texte** (suppression des accents, stemmatisation, lematisation, suppression des majuscules ...) et des **paramètres de la vectorisation** (taille du vocabulaire, filtres de fréquence, méthode de tokenization ...)

TP1 : Les méthodes de vectorisation statistiques : Bag-Of-Word, TF-IDF sont très sensibles à la définition du vocabulaire et aux pré-traitements.



Impact des prétraitements (stemmatisation, lematisation) sur la taille du vocabulaire

malgré un concept hautement cassé gueule la transcription de tubes rock récents en standards du swing paul anka nous délivre ici une leçon magistrale les reorchestrations sont inspirées et exécutées impeccablement anka fait preuve d'une grande maîtrise vocale alternant puissance dans les parties rythmées et caresse dans les temps lents nous rappelle au passage qu'il est après la disparition de frank sinatra pour lequel il avait écrit my way un des derniers grands crooneurs de notre époque sans oublier tony bennett quand même pas de faute de go dans ce cd qui voit l'interprète de diana et de put your head on my shoulder reprendre sans complexe du nirvana du van halen et de l'oasis le résultat est si stupefiant que l'on se demande qu'elle est finalement la version originale bien plus qu'un exercice de style rock swings est un vrai bon disque susceptible de plaire à tous les publics jazz et rock confondus chapeau bas

malgré un concept hautement cassé gueule la transcription de tubes rock récents en standards du swing paul anka nous délivre ici une leçon magistrale les reorchestrations sont inspirées et exécutées impeccablement anka fait preuve d'une grande maîtrise vocale alternant puissance dans les parties rythmées et caresse dans les temps lents nous rappelle au passage qu'il est après la disparition de frank sinatra pour lequel il avait écrit my way un des derniers grands crooneurs de notre époque sans oublier tony bennett quand même pas de faute de go dans ce cd qui voit l'interprète de diana et de put your head on my shoulder reprendre sans complexe du nirvana du van halen et de l'oasis le résultat est si stupefiant que l'on se demande qu'elle est finalement la version originale bien plus qu'un exercice de style rock swings est un vrai bon disque susceptible de plaire à tous les publics jazz et rock confondus chapeau bas

Visualisation des pondérations BoW (haut) et TF-IDF (bas)



Exploration de topics

Exploration de Topics

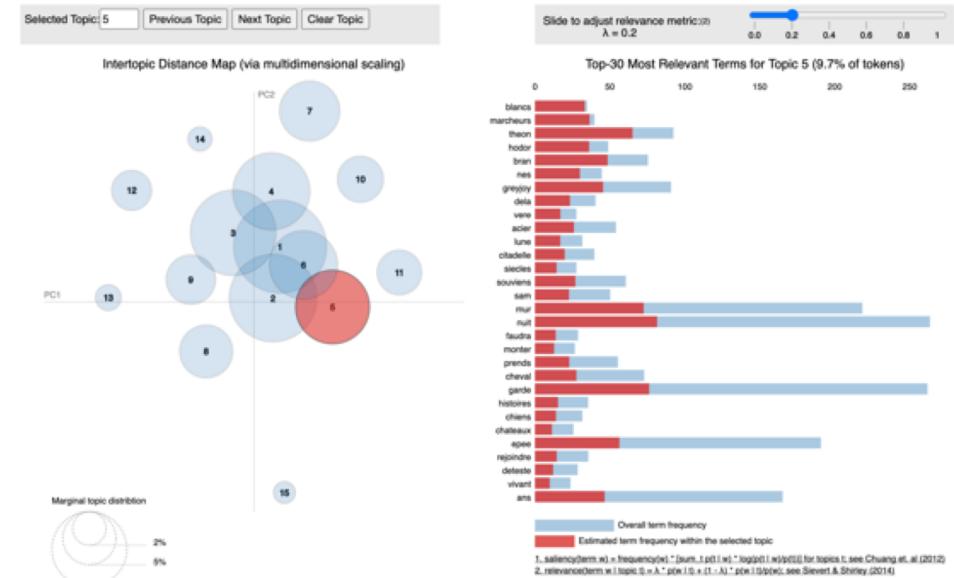
En traitement automatique du langage, on a souvent accès à **d'importants corpus mais souvent sans label** (par exemple des flux twitter)

L'exploration de topics permet de faire ressortir les thèmes récurrents dans un corpus. Nous avons en particulier détaillé l'algorithme de **Latent Dirichlet Allocation (LDA)** qui introduit une variable latente : les topics. Les documents sont alors représentés comme une **distribution sur les topics**. Les topics sont eux mêmes représentés comme une distribution sur les mots. Ceci suppose **d'interpréter les topics** en fonction de la forme de la distribution.

TP2 : La LDA, un algorithme d'exploration de topic très populaire. On introduit une variable latente : les topics. Les documents sont exprimés comme une distribution de topics et les topics comme une distribution de mots.



Variable latente dans le modèle LDA



Package de visualisation PyLDAVis



Les Embeddings de Mots



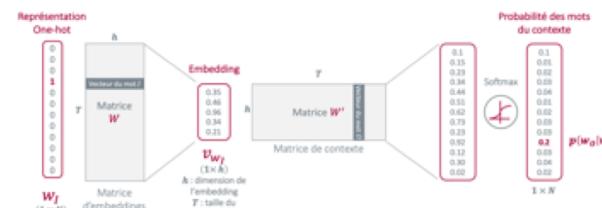
Les Embeddings de Mots

Les *words embeddings* désignent des représentations vectorielles des mots. Ces dernières sont en particulier de **faibles dimensions, denses et capturent les propriétés sémantiques des mots**.

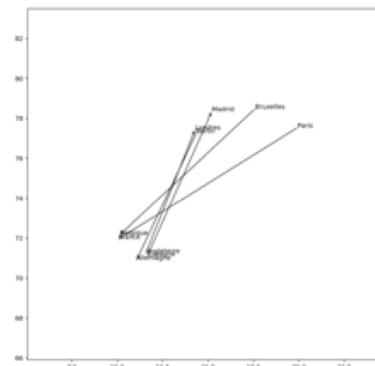
Les *words embeddings* sont appris de manières **auto-supervisée** : nous avons vu en particulier l'algorithme **Word2Vec** où l'on cherche à **prédirer un mot en fonction de son contexte** $p(w_{t+j}|w_t)$. Les matrices d'embeddings apprises lors de cette tâche peuvent ensuite être réutilisées pour d'autres applications

Les *words embeddings* peuvent être utilisés pour représenter des phrases. Nous avons vu le **modèle Bag-of-Word** où l'on somme les embeddings des mots de la phrase.

Cours 3 et TP4 : Les embeddings de mots. Exploration des vecteurs de représentation et un exemple d'utilisation pour l'analyse de sentiments.



L'architecture du modèle Word2Vec



Visualisation des relations Pays/Capitales



Architecture du modèle Bag-Of-Word pour les embeddings : on sommes les embeddings des mots de la phrase



Les modèles de langues et la génération de texte



Les modèles de langues et la génération de texte

Les **modèles de langue** (LM) cherchent à **estimer la probabilité des séquences de mots**. Ce type de modèles peuvent être utilisés pour la **reconnaissance de la parole**, la **correction des fautes d'orthographe**, l'**auto complétion de phrase** ou la **génération de texte**.

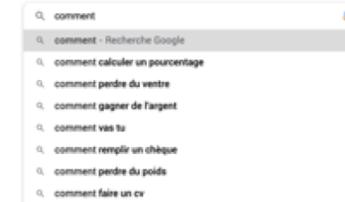
Dans les modèles n-grams, les mots de la phrase dépendent des n précédents. Par exemple dans le modèle bi-gram, la probabilité d'une phrase s'exprime $\mathbb{P}(W) = \mathbb{P}(w_1, w_2 \dots w_n) = \prod_{i=1}^n \mathbb{P}(w_i | w_{i-1})$

On évalue les probabilités des mots en fonction de la fréquence d'occurrence des n-grams. Par exemple dans le modèle trigram, $\mathbb{P}(w_3 | w_2, w_1) = \frac{c(w_1, w_2, w_3)}{c(w_1, w_2)}$ avec $c(w_1, w_2, w_3)$ la fréquence d'apparition du motif.

On peut évaluer la qualité d'un modèle de langue en fonction de la perplexité qui s'exprime : $\mathbb{P}\mathbb{P}(W) = \mathbb{P}(w_1, w_2 \dots w_n)^{-\frac{1}{n}}$

Finalement les modèles neuronaux permettent de tenir compte de l'ensemble des mots précédents : $\mathbb{P}(W) = \mathbb{P}(w_1, w_2 \dots w_n) = \prod_{i=1}^n \mathbb{P}(w_i | w_{i-1} \dots w_1)$. Nous avons utilisé le modèle GPT-2 pour la génération de texte.

Cours 4 : Les modèles de langues pour l'entraînement des modèles. Application à la génération de texte



Auto complétion dans Google

Dans son mouvement, la queue du Basilic lui avait jeté le Choixpeau magique à la tête. —Qu'est-ce qui se passe? demanda-t-il d'une voix aiguë.
 —Quoi?— démarre aussitôt de la foule, les yeux fixés sur le visage de Malefoy qui avait l'air de plus en plus livide, comme s'il n'avait pas eu le temps de prononcer le moindre mot... Harry se précipita dans la salle commune des Gryffondor, à côté de Ron et de Hermione, mais il ne fut pas surpris de voir que le professeur McGonagall était en train de dire quelque chose sur la Chambre des Secrets et qu'elle ne semblait pas convaincue que c'était la meilleure chose à faire...
 Il y eut un long silence, puis il se tourna vers Ron, le regard perdu dans ses pensées, et le silence qui régnait autour de lui se répercuta en

Génération de texte avec GPT-2 pour la fin de Harry Potter



Les modèles de Deep Learning



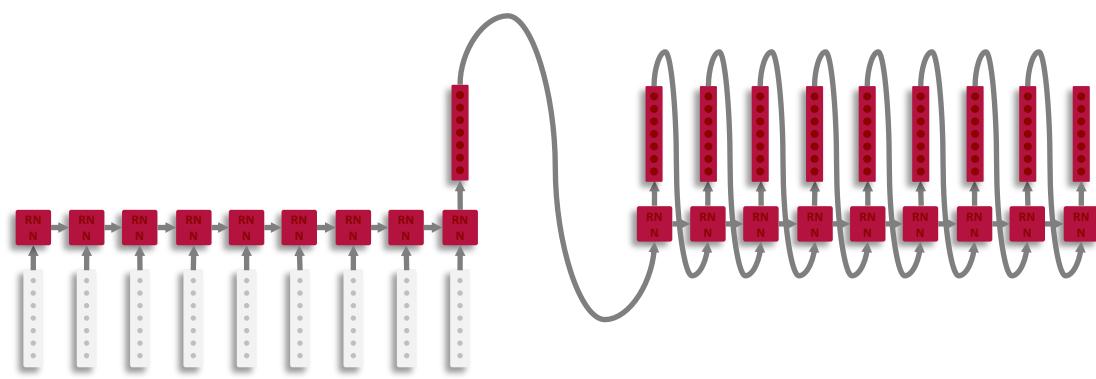
Les modèles de Deep Learning

Le **Deep Learning** est une branche du machine learning qui regroupe des algorithmes spécifiques : les **réseaux de neurones**. Les paramètres de ces architectures sont appris par **back-propagation**. Il existe de nombreuses spécificités du Deep Learning, l'aspect qui nous intéresse ici c'est que ce type de méthode présente généralement un pipeline simplifié ou les **features et les méthodes font partie de la même architecture et sont appris conjointement**.

Les **réseaux de neurones récurrents** modélisent les phrases comme des **séquences d'embeddings de mots**. Ils traitent l'entrée séquentiellement. A chaque étape, le vecteur de sortie est calculé en **fonction de l'embedding du mot courant et de l'état caché précédent**.

Les architectures transformers ont inspiré le modèle **BERT** qui consiste en une **succession de transformers**. Ce modèle est présent également la spécificité d'être **pré-entraîné** : les poids sont initialisés en entraînant le modèle sur une tâche auto-supervisée. En l'occurrence prédire des mots masqués, on parle de modèle de langue masqué. Ce modèle est très versatile et **peut être décliné pour quasiment tous les cas d'usages du NLP**.

Cours 5 : Les modèles de Deep Learning. Catalogue des modèles état de l'art pour le NLP.



Architecture Sequence-to-Sequence avec réseaux récurrents



Les systèmes Q/R



Les systèmes
Q/R

Les systèmes de Questions/Réponses sont des méthodes supervisées, on concatène une question et un paragraphe qui contient la réponse. Le modèle est entraîné à produire pour chaque token sa probabilité de correspondre au début de la réponse ou à la fin de la réponse. On identifie le passage de plus haute probabilité comme la réponse. Une fonction de loss permet de comparer la sortie à la réponse préalablement identifiée.

Cours 5 : Utilisation de Bert pour un système de Questions/Réponses.

```
Entrée [115]: 1 question = "Quand est venu au monde Paul Jules Antoine Meillet ?"
2
3
4 context = """Paul Jules Antoine Meillet, né le 11 novembre 1866 à Moulins (Allier) et mort
5 le 21 septembre 1936 à Châteaumeillant (Cher), est le principal linguiste français des
6 premières décennies du xxe siècle. Il est aussi philologue. D'origine bourbonnaise, fils
7 d'un notaire de Châteaumeillant (Cher), Antoine Meillet fait ses études secondaires au lycée
8 de Moulins. Étudiant à la faculté des lettres de Paris à partir de 1885 où il suit notamment
9 les cours de Louis Havet, il assiste également à ceux de Michel Bréal au Collège de France et
10 de Ferdinand de Saussure à l'École pratique des hautes études."""
11
12
13 # 1. On tokenize l'input
14 inputs = tokenizer.encode_plus(question, context, return_tensors="pt")
15
16 # 2. On prédit avec le modèle l'index de début et de fin de la réponse dans le texte
17 answer_start_scores, answer_end_scores = model_qa(**inputs)
18 answer_start = torch.argmax(answer_start_scores)
19 answer_end = torch.argmax(answer_end_scores) + 1
20
21 # 3. On renvoie le span de texte avec la réponse
22 tokenizer.convert_tokens_to_string(tokenizer.convert_ids_to_tokens(inputs["input_ids"])[0][answer_start:answer_end])

Out[115]: '11 novembre 1866'

Entrée [116]: 1 color_answer_in_text(question, context, answer_start, answer_end)

Paul Jules Antoine Meillet, né le 11 novembre 1866 à Moulins (Allier) et mort le 21 septembre 1936 à Châteaumeillant (Cher), est le principal linguiste français des premières décennies du xxe siècle. Il est aussi philologue. D'origine bourbonnaise, fils d'un notaire de Châteaumeillant (Cher), Antoine Meillet fait ses études secondaires au lycée de Moulins. Étudiant à la faculté des lettres de Paris à partir de 1885 où il suit notamment les cours de Louis Havet, il assiste également à ceux de Michel Bréal au Collège de France et de Ferdinand de Saussure à l'École pratique des hautes études.
```

Modèle de Questions/Réponses testé sur de
exemples de Wikipédias



Les librairies utilisées

NLP

spaCy

spacy.io

Très efficace pour toutes les prétraitements et les méthodes d'analyse linguistique. Des modèles en français. Une implémentation très efficace.

NLTK

nltk.org

Librairie un peu vieillissante de NLP. Encore assez efficace pour certains prétraitements.

scikit-learn

scikit-learn.org/

L'indétrônable librairie française de ML. L'organisation des classes toujours très pratique. Pour le NLP, le TF-IDF est très efficace.



docs.python.org/3/library/re.html

La librairie **re** de python : très utile pour les prétraitements

Deep Learning

GENSIM

radimrehurek.com/gensim/

Un outil qui n'a pas son pareil pour entraîner des embeddings.



huggingface.co

Une bibliothèque pour tous les modèles les plus récents de NLP. Mise à jour très régulièrement.

TensorFlow

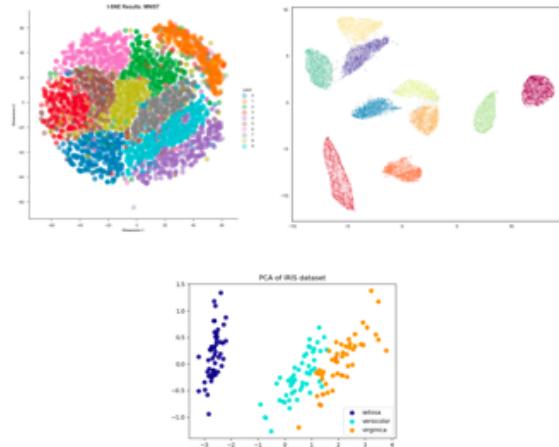
tensorflow.org

La grosse librairie de Deep Learning avec Tensorflow. Les deux ont leurs avantages et leurs inconvénients.

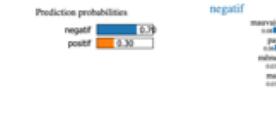
Les outils de visualisations et d'intélligibilité



Tensorboard : un outil de visualisation assez complet



t-SNE, UMAP, PCA sont trois méthodes de réduction de dimension. Elles permettent notamment de projeter les données en 2D pour les visualiser.



Text with highlighted words
Les bons points : - des faits vérifiables ; tout s'enchaîne logiquement. Des détails sont bien cachés. On le voit plus clair sur certaines choses bien cachées. Les mauvais points : - l'auteur est un fâche. Ne pensez surtout pas que c'est un banquier (quel banquier irait écrire un livre expliquant les résultats d'un vol scandaleux que lui même a fait dans la partie...). Cela n'a rien à faire d'un multidimensionnel capable d'insulter ses supérieurs ? Passons toutes les grossièretés qui trahissent clairement que le journaliste (oui, j'insiste) : un banquier, ne soyons pas dupes) qui a écrit cela pour se agir par la haute société et leurs pratiques détestables. Cela n'a rien à faire d'un journaliste accepté tout simplement parce que c'est eux qui ont le pouvoir. Bref, un livre pour s'inscrire sur l'histoire cachée des riches aussi ou sont leur de même enrichis sur le dos de la

Lime (Túlio Ribeiro et al., 2016) est une méthode d'analyse des modèles qui cherche à expliciter sur quelles caractéristiques de l'échantillon en entrée, l'algorithme s'est appuyé pour formuler sa prédition.