

Natural Language Processing

Redha Moulla
Novembre 2022

Plan de la formation

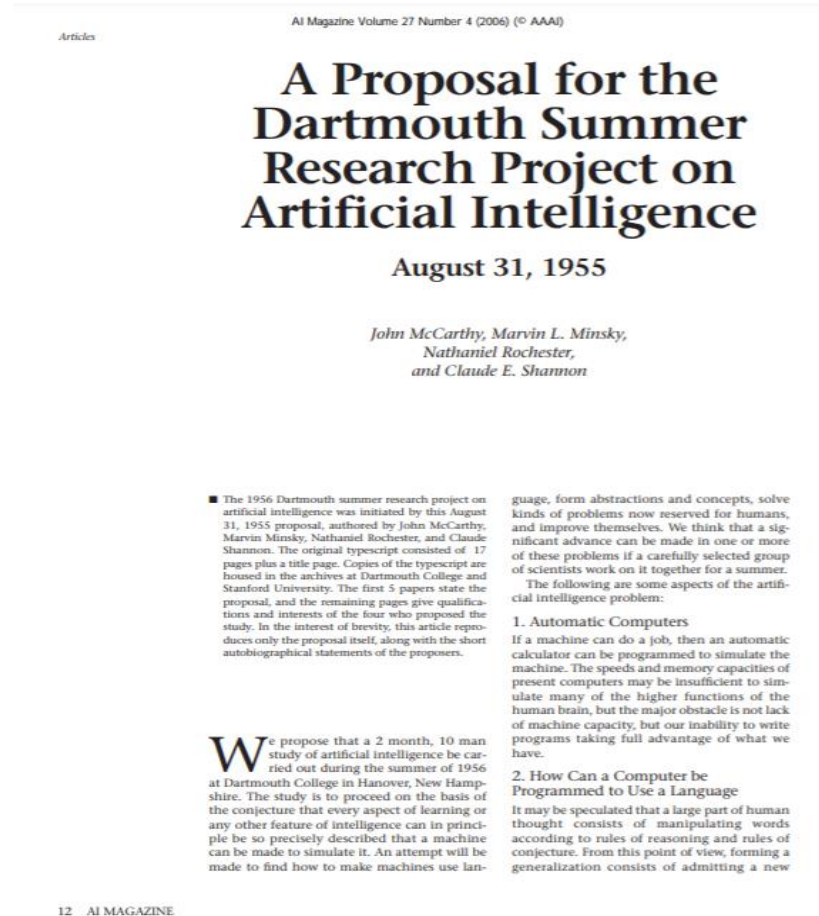
- Introduction au NLP
- Techniques statistiques pour le NLP
- Machine learning pour le NLP
- Introduction au plongement lexical.
- Deep learning pour le NLP
- Transformers et apprentissage auto-supervisé.
- Apprentissage multimodal.

Principaux outils

- Google Colab
- Python 3
- Bibliothèques classiques de NLP : NLTK, Spacy
- Pytorch

1956 : conférence de Dartmouth

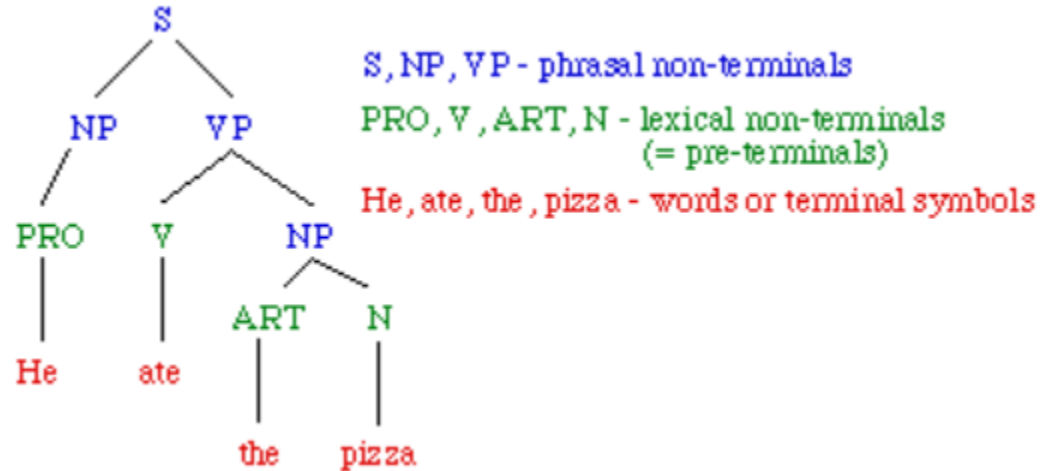
L'histoire du NLP est intimement liée à celle de l'intelligence artificielle. Il figure même dans le programme de la conférence de Dartmouth, qui a fondé l'IA.



“We propose that a 2-month, 10-man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.”

1970: approches traditionnelles du NLP

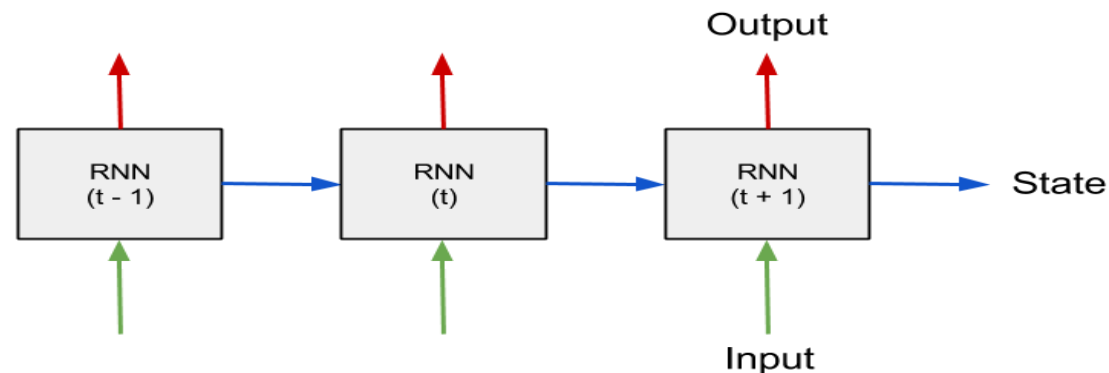
Les modèles traditionnels reposent majoritairement sur une approche linguistique. Ils consistent généralement à « parser » le langage naturel, même quand celui-ci possède une structure complexe.



1986: Les réseaux de neurons récurrents

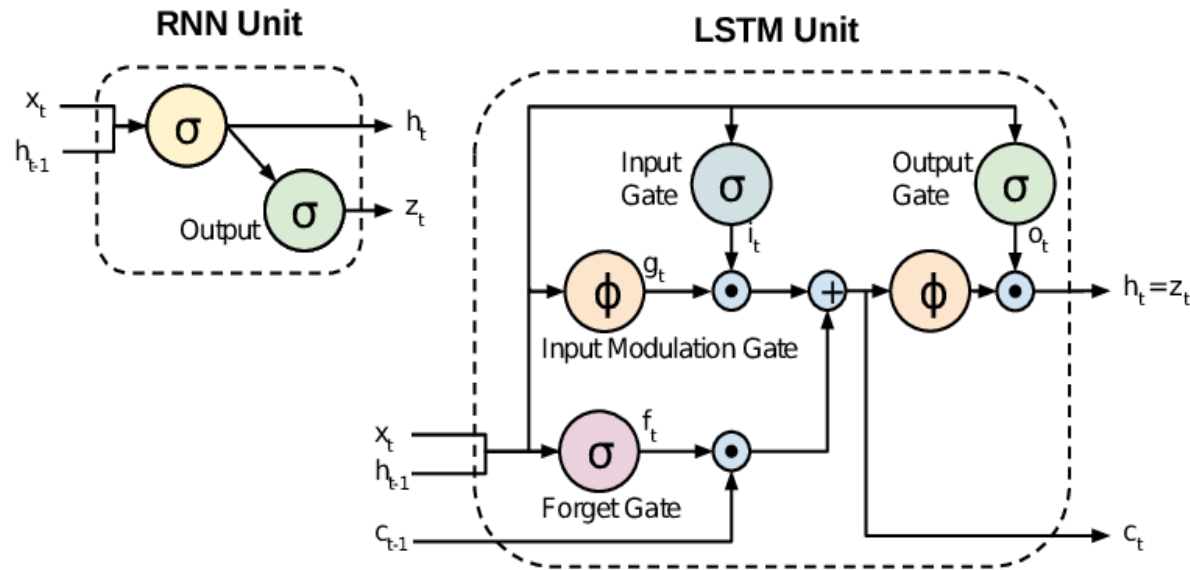
Les RNNs permettent de traiter des séquences ; ils sont ainsi naturellement adaptés au langage naturel. Ils ont cependant rapidement montré des limites.

We describe a new learning procedure, back-propagation, for networks of neurone-like units. The procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. As a result of the weight adjustments, internal 'hidden' units which are not part of the input or output come to represent important features of the task domain, and the regularities in the task are captured by the interactions of these units. The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure¹.



1997: Long Short-Term Memory

Les LSTMs sont nés pour dépasser les limites des RNNS, en particulier le problème de l'annulation du gradient.



2010s: Renaissance du deep learning

Plusieurs avancées ont été réalisées durant les dernière décennie en deep learning, notamment depuis 2012 (AlexNet).

- 2013 : représentations distribuées et Word2Vec
- 2014 : Emergence du mécanisme de self-attention
- 2017 : Emergence des modèles Transformers
- 2018: ELMo, BERT, GPT.
- 2020: GPT-3, etc.

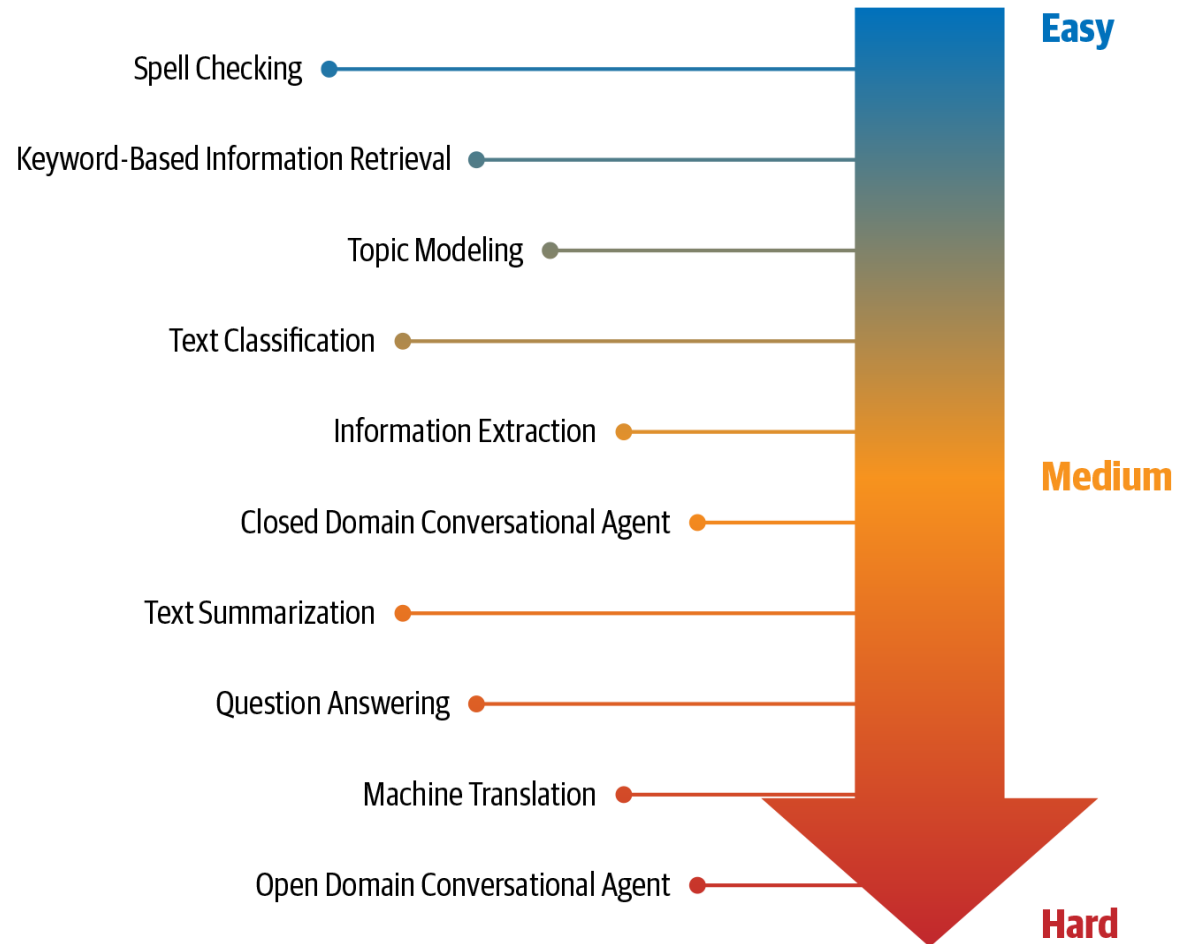
Tâches classiques en NLP

On retrouve dans le NLP un certain nombre de tâches classiques, qui servent à la fois dans les applications pratiques que dans l'évaluation de l'état de l'art.

- Modélisation du langage (language modeling)
- Classification de texte
- Extraction d'informations
- Détection de thématiques (topic modeling)
- Résumé de texte
- Question/réponse
- Etc.

Classement des tâches en NLP par ordre de difficulté

Le classement ci-dessous est à titre indicatif et sujet à des changements. Certaines tâches, comme la traduction, paraissent aujourd'hui moins difficiles qu'il y a quelques années.



Pourquoi le NLP est si difficile ?

- **Le langage naturel est ambigu**
“Acheter un avocat”
- **Le langage naturel repose sur le sens commun**
Il fait froid en hiver
Le beurre fond dans le four
- **Le langage naturel n’est pas basé sur des règles**
*En bleu adorable fleurit
Le toit de métal du clocher. Alentour
Plane un cri d’hirondelles, autour
S’étend le bleu le plus touchant. Le soleil*
Friedrich Hölderlin

Pré-traitement des données textuelles

Les données textuelles sont non structurées. Les principales étapes de pré-traitement impliquent :

1. Segmentation des phrases et tokénisation ;
2. Suppression de stop words, de la ponctuation, etc.
3. Stemming, lemmatisation, conversion des majuscules, etc.

Et

Détection du langage, POS tagging, etc.

Tokénisation

La tokénisation consiste à segmenter une phrase en unités plus petites (tokens), qui sont généralement des mots, mais qui peuvent également être des caractères ou ensembles de caractères.

"Il fait beau aujourd'hui à Paris" → ["Il", "fait", "beau", "aujourd'hui", "à", "Paris"]

"Paris" → ["P", "a", "r", "i", "s"]

La tokénisation par subword est utilisée d'une manière assez fréquente en deep learning.

Stemming and lemmatisation

- Le stemming est l'opération qui consiste à supprimer les suffixes et réduire les mots à leur forme de base.

```
marcher --> march  
marchait --> march  
marcha --> march  
marchassiez --> march  
marché --> march
```

- La lemmatisation est l'opération qui consiste à réduire un mot à sa racine.

```
nous --> nous  
sommes --> être  
venus --> venir  
faire --> faire  
les --> le  
marchés --> marché
```

Vectorisation

- One-Hot encoding: Etant donné un vocabulaire V , chaque mot est représenté par un vecteur binaire (à 0 ou 1) de dimension V .

$V = \{\text{Tom, mange, pomme, chien, ciel}\}, \quad \text{où } \dim(V) = 5$

Tom = [1, 0, 0, 0, 0]

mange = [0, 1, 0, 0, 0]

pomme = [0, 0, 1, 0, 0]

etc.

- N-grams

“machine learning est très utilisé en NLP.”

1-gram: {machine, learning, est, très, utilisé, en, NLP}

2-gram: {machine learning, learning est, est utilisé, utilisé en, en NLP}

Extraction d'informations

L'information peut concerner différentes entités (événements clés, personnes, lieux, etc.).

L'extraction d'information inclut :

- Détection d'entités nommées (NER).
- Extraction de mots clés (KPE).
- Extraction de relations.

Entités nommées (NER)

L'extraction d'entités nommées consiste à détecter des informations clés qui sont des noms propres (noms de personnes, de lieux, etc.)

Established in 1896, the site was taken over by **Askham Bryan** in 2011 and it has 536 **students**, including apprentices. A group set up to keep it open had tried to find another college to take it over, after two previous bids were deemed unsuitable. Tim Whitaker, chief executive officer and principal at Askham Bryan College, said he regretted "the upset" the closure and job losses will cause.

"Whilst it was very disappointing that the strategic review didn't receive a sustainable option for Newton Rigg campus, we welcome the plans for the preservation of land-based provision in Cumbria.

"We will support and work with those involved in these plans, to ensure that current students and future applicants interested in land-based courses have a smooth transition."

He added that the college had "always been clear" educational provision would not continue at Newton Rigg from July 2021, and students, staff and the local community were told in 2020.

Extraction de relations

L'extraction de relations consiste à détecter des relations entre deux entités nommées.

Established in 1896, the site was taken over by Askham Bryan in 2011 and it has 536 students, including apprentices.

A group set up to keep it open had tried to find another college to take it over, after two previous bids were deemed unsuitable.

Tim Whitaker, chief executive officer and principal at Askham Bryan College, said he regretted "the upset" the closure and job losses will cause.

"Whilst it was very disappointing that the strategic review didn't receive a sustainable option for Newton Rigg campus, we welcome the plans for the preservation of land-based provision in Cumbria.

"We will support and work with those involved in these plans, to ensure that current students and future applicants interested in land-based courses have a smooth transition."

He added that the college had "always been clear" educational provision would not continue at Newton Rigg from July 2021, and students, staff and the local community were told in 2020.

Extraction de mots clés

L'extraction de mots clés consiste à extraire les mots clés les plus importants, permettant de capturer le contenu d'un texte.

Established in 1896, the site was taken over by Askham Bryan in 2011 and it has 536 **students**, including apprentices. A group set up to keep it open had tried to find another college to take it over, after two previous bids were deemed unsuitable. Tim Whitaker, chief executive officer and principal at Askham Bryan College, said he regretted "the upset" the closure and job losses will cause.

"Whilst it was very disappointing that the strategic review didn't receive a sustainable option for Newton Rigg campus, we welcome the plans for the preservation of land-based provision in Cumbria.

"We will support and work with those involved in these plans, to ensure that current students and future applicants interested in land-based courses have a smooth transition."

He added that the college had "always been clear" educational provision would not continue at Newton Rigg from July 2021, and students, staff and the local community were told in 2020.

Part of Speech tagging

Part of speech consiste à détecter la catégorie grammaticale d'un mot au sein d'un texte.

Exemple :

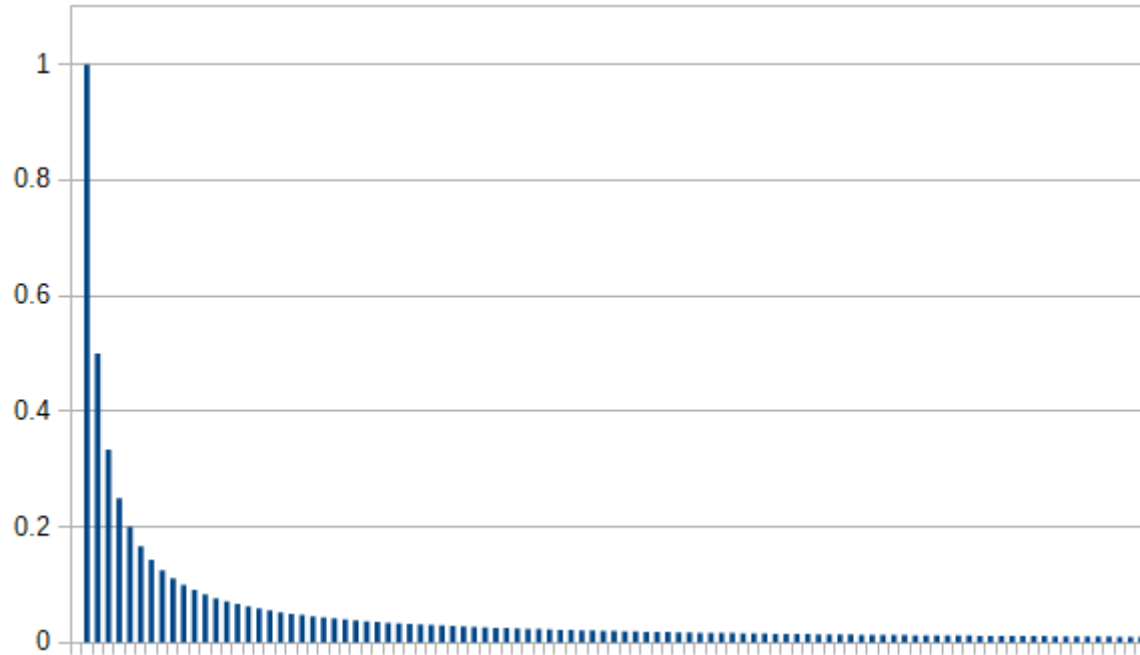
« this is a very simple example »

```
[('this', 'DT'), ('is', 'VBZ'), ('a', 'DT'), ('very', 'RB'), ('simple', 'JJ'), ('example', 'NN')]
```

Techniques statistiques pour le NLP

Principe des techniques statistiques pour le NLP

- Un document est caractérisé par la distribution des mots qui le composent.
- La distribution des mots obéit généralement à une loi de puissance (ou de Zipf). La majorité du langage que nous utilisons ordinairement est composé d'un nombre très limité de mots.



Term Frequency

- Term Frequency (TF) : la fréquence avec laquelle un terme t apparaît dans un document d .

$$\text{TF}(t, d) = \frac{(\text{Number of occurrences of term } t \text{ in document } d)}{(\text{Total number of terms in the document } d)}$$

Exemple : étant donné 1000 mots.

Terme	Fréquence	TF score
chat	15	0.015
lait	4	0.004
mange	6	0.006
feu	3	0.003
...

Inverse Document Frequency

- Inverse Document Frequency (IDF) : permet de pénaliser les termes les plus fréquents et de donner plus de poids aux termes les moins fréquents (plus parlants).

$$\text{IDF} \left(t \right) = \log_e \frac{(\text{Total number of documents in the corpus})}{(\text{Number of documents with term } t \text{ in them})}$$

- Example: given 100 documents

Terme	Nombre de documents	IDF score
chat	70	0.36
lait	9	2.41
manger	80	0.22
feu	5	3.00
...

Term Frequency – Inverse Document Frequency

- TF-IDF score est le produit de Tf et IDF.

$$TF\text{-}IDF\text{ Score} = TF\text{ Score} \times IDF\text{ score}$$

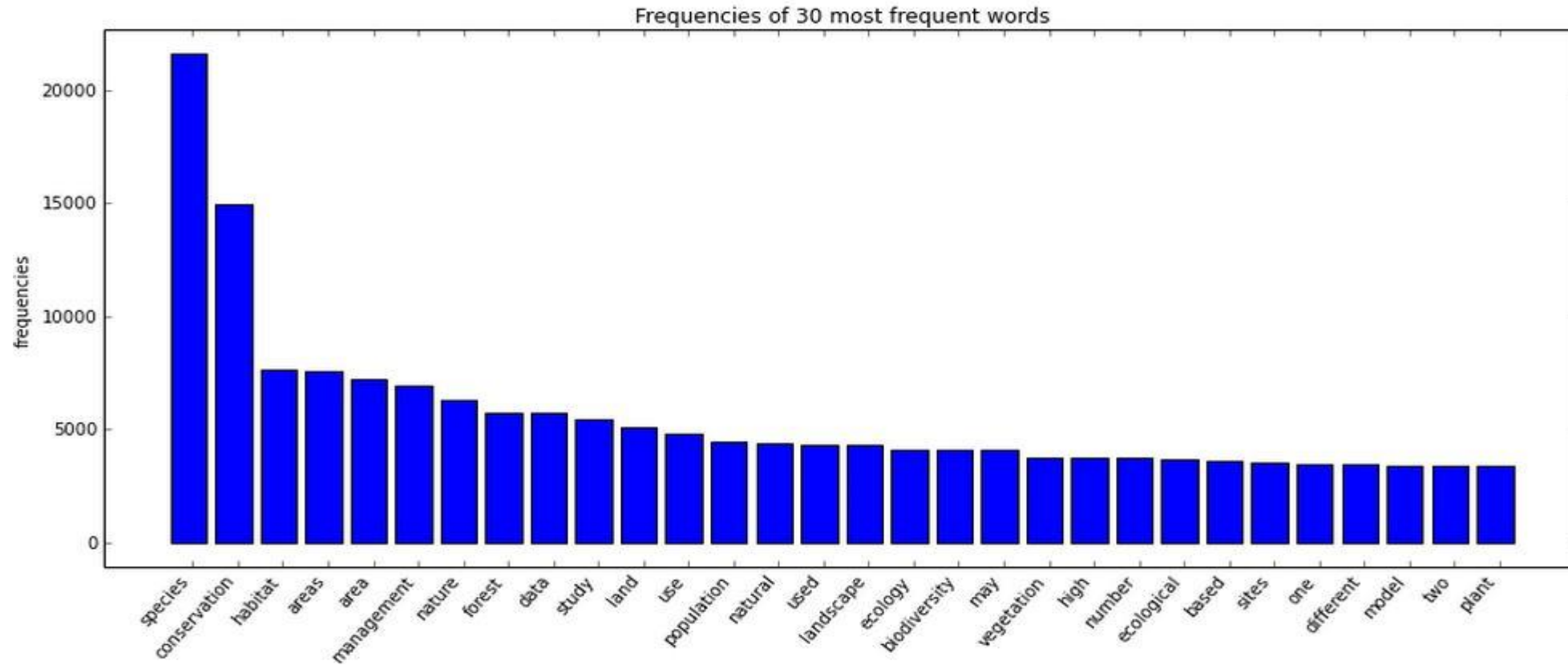
Terme	TF score	IDF score	TF-IDF score
chat	0.015	0.36	0.0054
lait	0.004	2.41	0.0096
manger	0.006	0.22	0.0013
feu	0.003	3.00	0.0090
...	

Topic Modeling

- *Topic modeling is a technique for automatically organizing, understanding, searching and summarizing large electronic archives.* (David Blei).
- Découvrir des thématiques sous-jacentes à un document ou un ensemble de documents (Google News, etc.)
- Annoter ou caractériser les documents selon les thématiques.

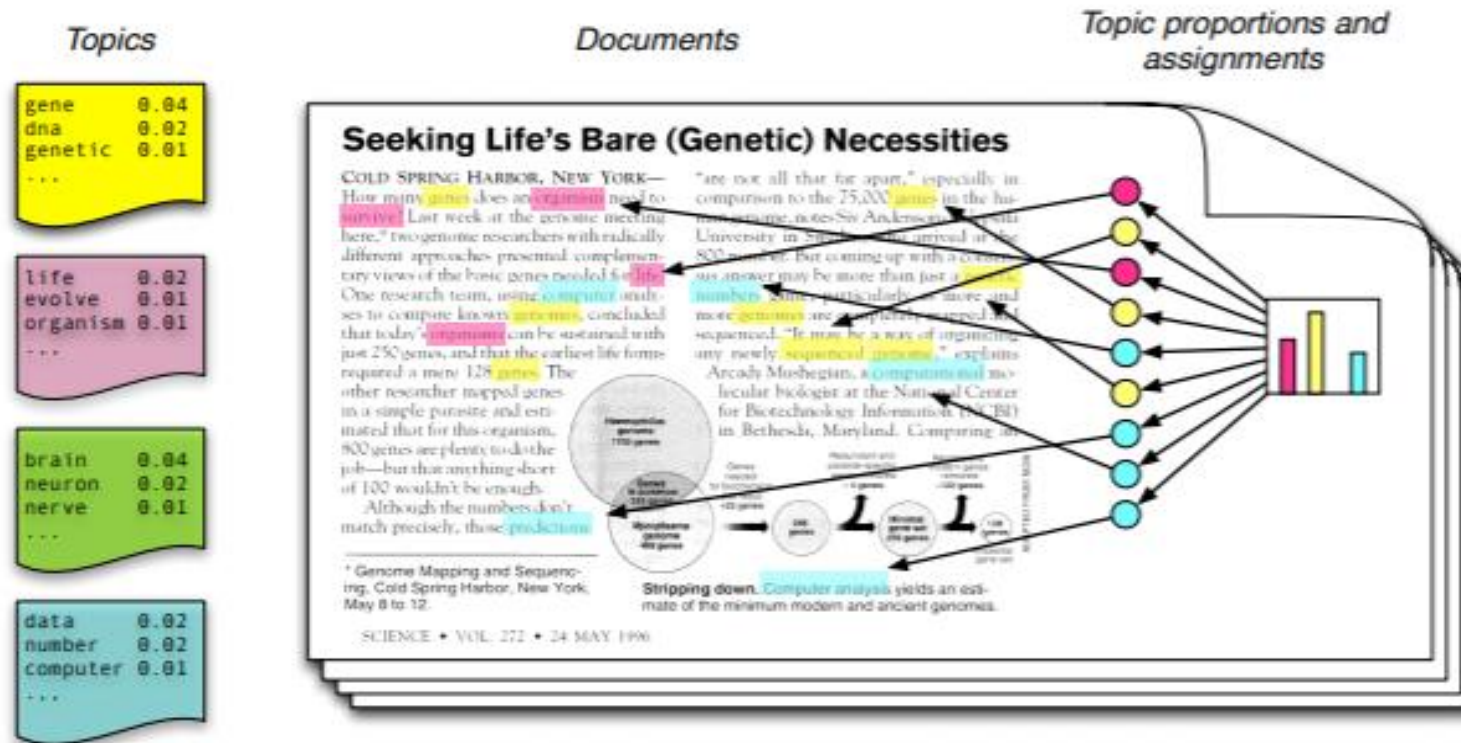
Topic Modeling : thématiques

- Une thématique est caractérisée par une distribution de mots (la fréquence de certains mots).



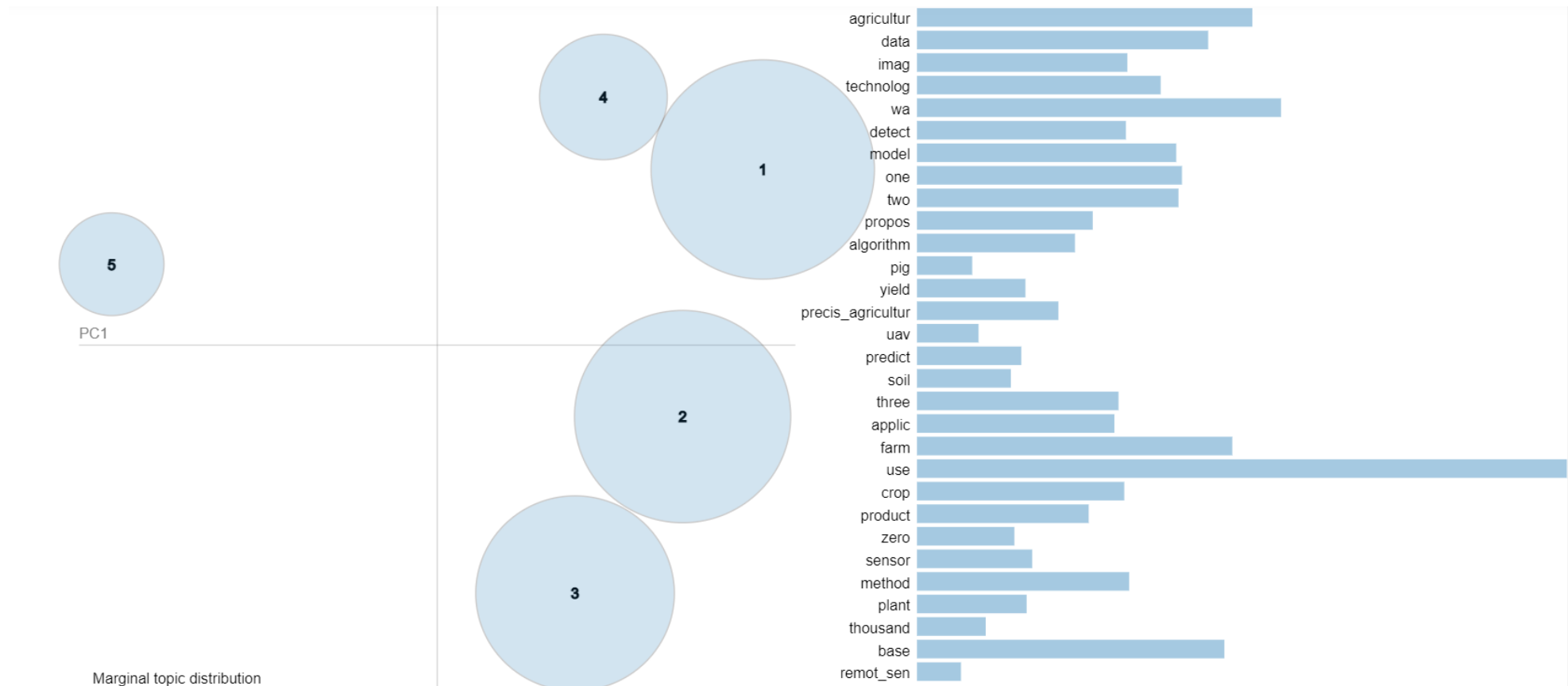
Topic Modeling : documents

Un document est d'une manière générale un mélange de thématiques.



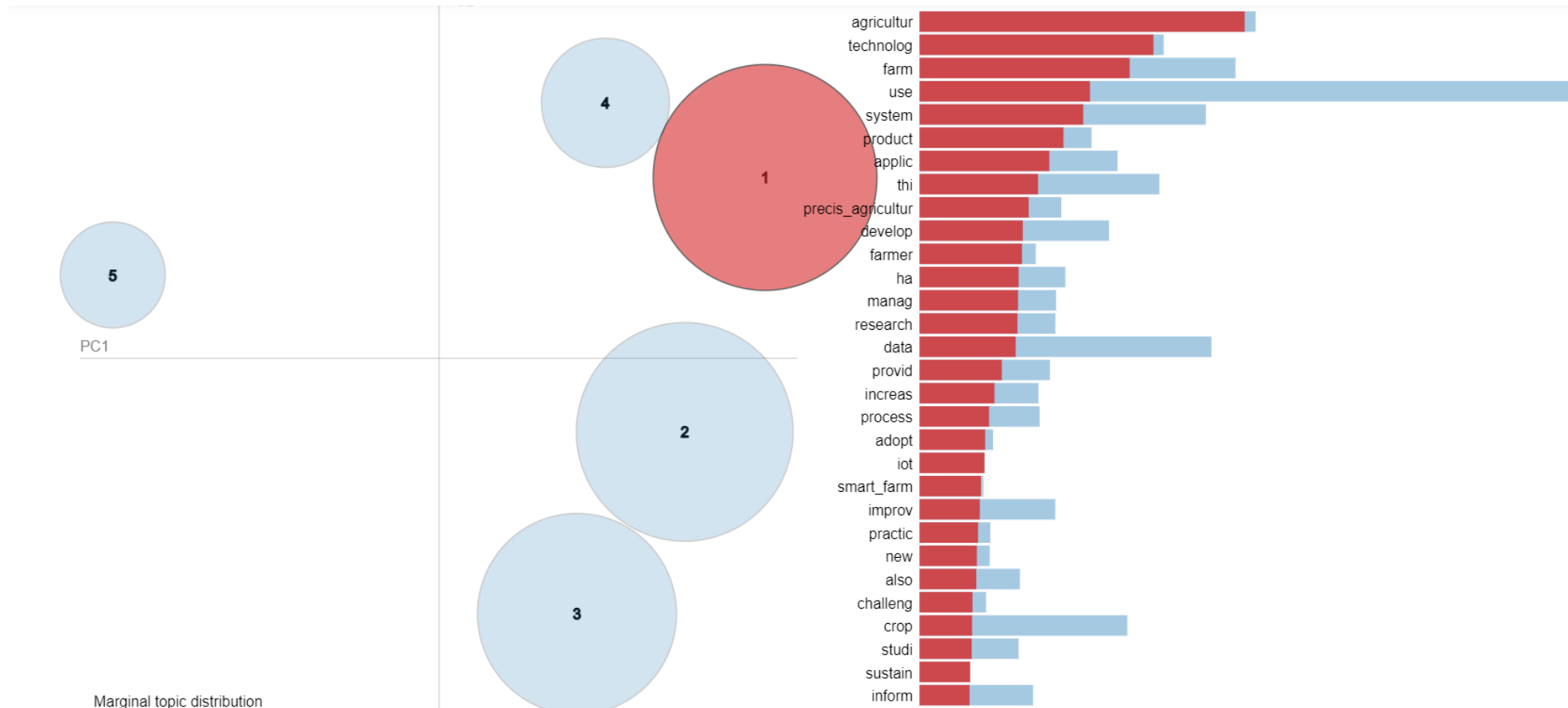
Topic Modeling : LDA

Extraction de thématiques d'un ensemble de documents traitant de l'agriculture l'aide d'une technique LDA (Latent Dirichlet Allocation).



Topic Modeling : exemple d'une thématique

La thématique extraite est définie par un ensemble de mots clés.



Latent Dirichlet Allocation (LDA)

Nous disposons d'une collection de documents dont nous voulons identifier les thématiques.

Doc1	Doc2	Doc3	Doc4
Maïs	Roman	Véhicule	Tracteur
Blé	Lecteur	Moteur	Culture
Maïs	Lecteur	Moteur	Moteur
Ferme	Editeur	Tracteur	Moteur
Culture	Culture	Puissance	Blé
Blé	Roman	Roman	Arbre
Tracteur	Culture	Culture	Maïs

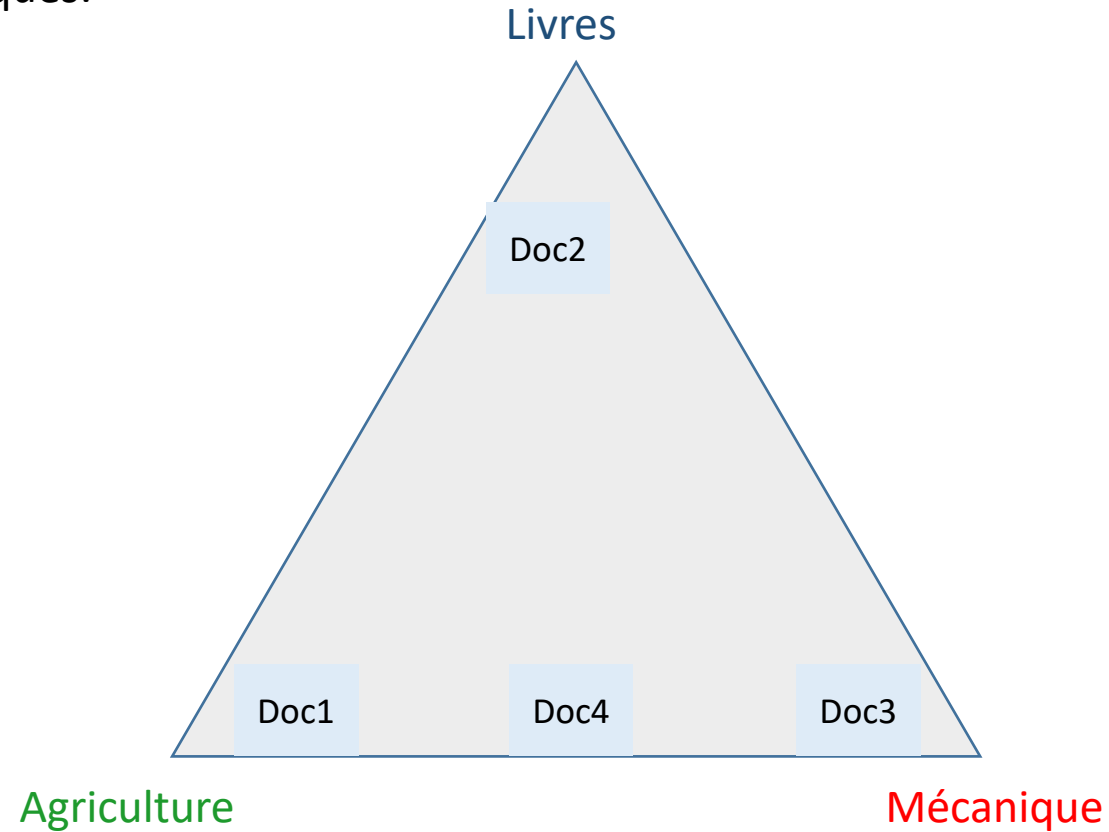
Agriculture

Livres

Mécanique

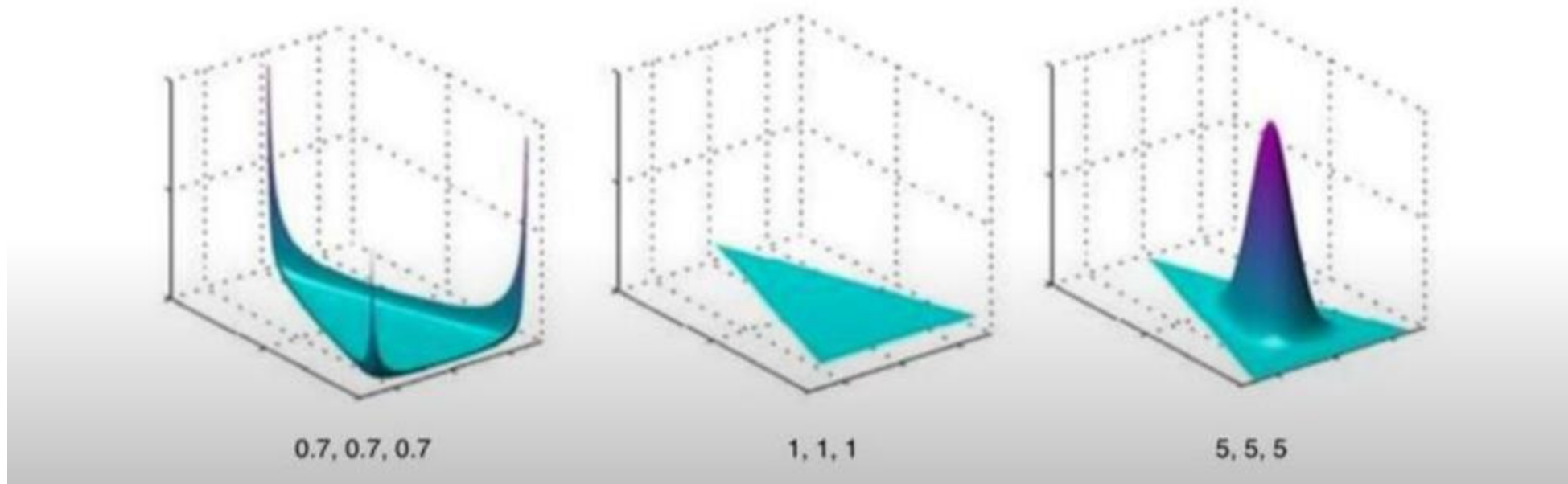
Latent Dirichlet Allocation (LDA)

L'algorithme LDA essaie d'assigner à chaque document une thématique ou, plus généralement, un ensemble de thématiques.



Distribution de Dirichlet

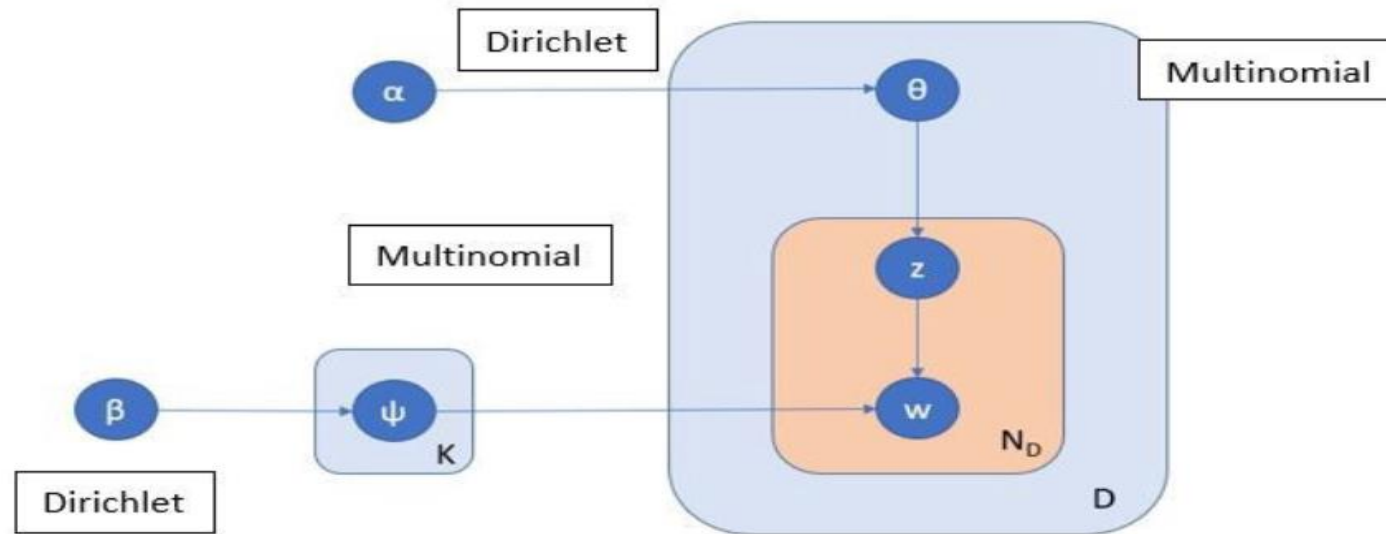
$$f(x_1, \dots, x_K; \alpha_1, \dots, \alpha_K) = \frac{1}{B(\alpha)} \prod_{i=1}^K x_i^{\alpha_i - 1}$$



LDA : un modèle génératif

L'algorithme LDA tente de générer les documents à partir du vocabulaire en estimant les paramètres du modèle suivant.

$$P(\mathbf{W}, \mathbf{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}; \alpha, \beta) = \prod_{j=1}^M P(\theta_j; \alpha) \prod_{i=1}^K P(\varphi_i; \beta) \prod_{t=1}^N P(Z_{j,t} | \theta_j) P(W_{j,t} | \varphi_{Z_{j,t}})$$



LDA : entraînement

Plus concrètement, l'algorithme LDA se décline selon les étapes suivantes :

- Associer les mots aux thématiques d'une manière aléatoire.

Pour chaque mot w d'un document d , calculer :

- La probabilité $p(t \mid d)$ que le document d soit associée à la thématique t .
- La probabilité $p(w \mid t)$ que la thématique t soit associée au mot w .
- On associe alors la thématique t au mot w du document d avec la probabilité $p(t \mid d) \times p(w \mid t)$.
- Répéter ces étapes jusqu'à convergence (stabilisation des associations des mots aux thématiques).

Latent Dirichlet Allocation (LDA)

Nous disposons d'une collection de documents dont nous voulons identifier les thématiques.

Doc1	Doc2	Doc3	Doc4
Maïs	Roman	Véhicule	Tracteur
Blé	Lecteur	Moteur	Culture
Maïs	Lecteur	Moteur	Moteur
Ferme	Editeur	Tracteur	Moteur
Culture	Culture	Puissance	Blé
Maïs	Roman	Roman	Arbre
Tracteur	Culture	Culture	Maïs

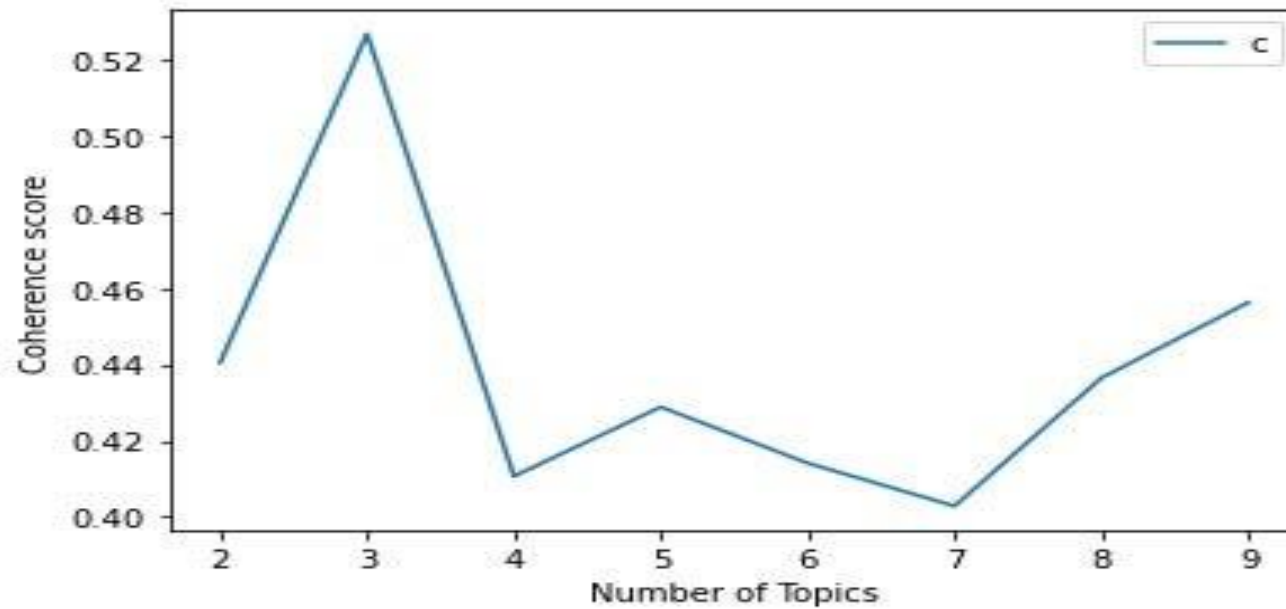
Latent Dirichlet Allocation (LDA)

Nous disposons d'une collection de documents dont nous voulons identifier les thématiques.

Doc1	Doc2	Doc3	Doc4
Maïs	Roman	Véhicule	Tracteur
Blé	Lecteur	Moteur	Culture
Maïs	Lecteur	Moteur	Moteur
Ferme	Editeur	Tracteur	Moteur
Culture	Culture	Puissance	Blé
Maïs	Roman	Roman	Arbre
Tracteur	Culture	Culture	Maïs

Topic Modeling : nombre optimal de thématiques

La cohérence de thématique peut être estimée à l'aide d'une métrique appelée cohérence score. Celle-ci définit le nombre optimal de thématiques associées au corpus de documents.



Topic Modeling : mesure de la cohérence des thématiques

La cohérence de thématique peut être estimée à l'aide d'une métrique appelée cohérence score. Celle-ci définit le nombre optimal de thématiques associées au corpus de documents.

- Le score U_Mass : il mesure la fréquence d'occurrence des mots deux à deux dans un document.

$$C_{\text{UMass}} = \frac{2}{N \cdot (N - 1)} \sum_{i=2}^N \sum_{j=1}^{i-1} \log \frac{P(w_i, w_j) + \epsilon}{P(w_j)}$$

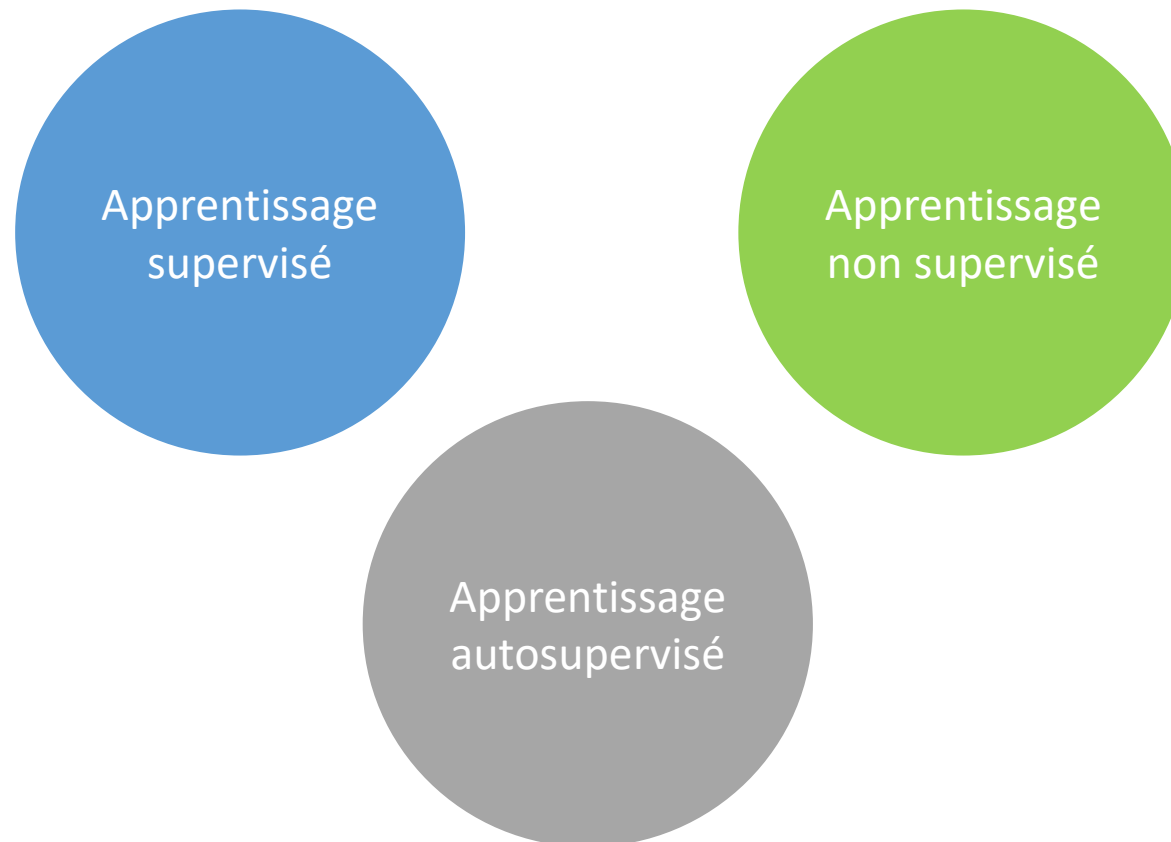
- Le score C_V :

$$v_{ij} = \text{NPMI}(w_i, w_j)^\gamma = \left(\frac{\log \frac{P(w_i, w_j) + \epsilon}{P(w_i) \cdot P(w_j)}}{-\log(P(w_i, w_j) + \epsilon)} \right)^\gamma$$

Machine learning pour le NLP

Introduction au machine learning

Trois grandes familles d'apprentissage



L'apprentissage supervisé

L'apprentissage supervisé consiste à apprendre une fonction qui associe une étiquette (label) à un ensemble de caractéristiques (features).

- Inputs : un jeu de données **annotées** pour entraîner le modèle.
Exemple : des textes (tweets, etc.) avec les sentiments associés, positifs ou négatifs.
- Output : une étiquette pour un point de donnée inconnu par le modèle.
- Une très large classe de modèles possibles (linéaire, arbres de décision, réseaux de neurones), chacun impliquant un ou un ensemble de biais inductifs.

L'apprentissage supervisé : deux familles

L'apprentissage supervisé se décline lui-même en deux grandes familles :

- La régression : prédire une valeur continue (un nombre réel typiquement).
Exemple : prédire le prix d'un appartement, la life time value d'un client, etc.
- La classification : prédire une catégorie ou une classe.
Exemple : prédire l'étiquette d'une image (chat, chien, etc.), le sentiment associé à un texte, le centre d'intérêt d'un client à partir de ses commentaires, etc.
- En NLP, nous sommes principalement concernés par la classification, mais on peut également extraire des features de données textuelles à des fins de régression.

Métriques associées à la classification

Les modèles de classification supervisée peuvent être évalués à l'aide de différentes métriques. Les plus communes sont :

$$\text{Précision} = \frac{vp}{vp+vn}$$

$$\text{Rappel} = \frac{vp}{vp+fn}$$

$$\text{F1 Score} = \frac{2PR}{P+R}$$

Concept de surapprentissage

On dit qu'un modèle surapprend (ou overfit) quand il est trop adapté aux données d'entraînement, ce qui induit une mauvaise généralisation aux données de test, a fortiori celle du monde réel.

Le surapprentissage apparaît notamment quand :

- Le nombre de points de données est très faible (grande dimension, etc.);
- Le modèle est trop complexe par rapport aux données.

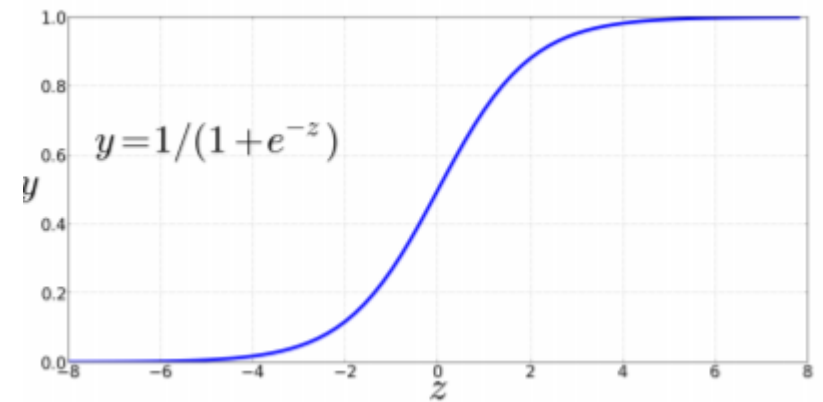
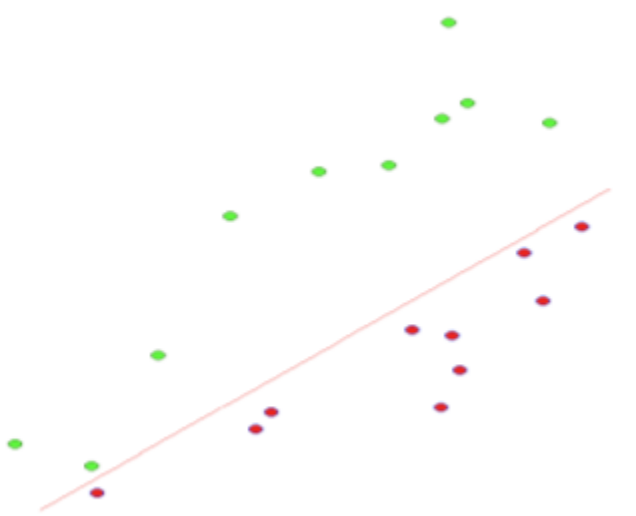
Régularisation

Il y a plusieurs approches ou techniques permettant de réduire le problème de surapprentissage.

- Des techniques de régularisation de type L1, L², dropout, etc.
- Construire un modèle plus simple
- Assembler plus de données pour l'entraînement.
- Etc.

Régression logistique

La régression essaie de séparer les données d'une manière linéaire.



Régression logistique

Avantages :

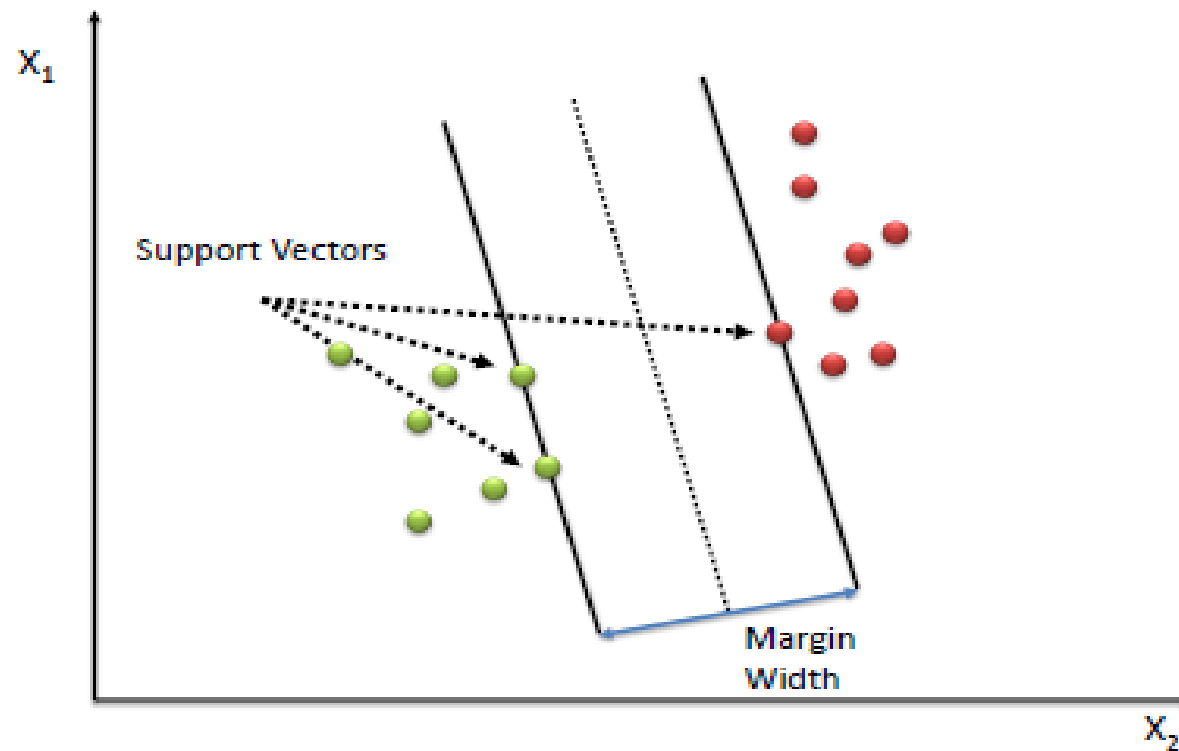
- Modèle simple et facile à expliquer.
- Le risque de surapprentissage est relativement limité.

Inconvénients :

- Modèle trop simple si les données sont complexes.
- Très peu robuste pour les données déséquilibrées.

Machines à vecteurs de support (SVM)

Les SVMs tentent de séparer les données en construisant la surface de séparation ayant la plus vaste marge entre deux classes.



Machines à vecteurs de support (SVM)

Avantages :

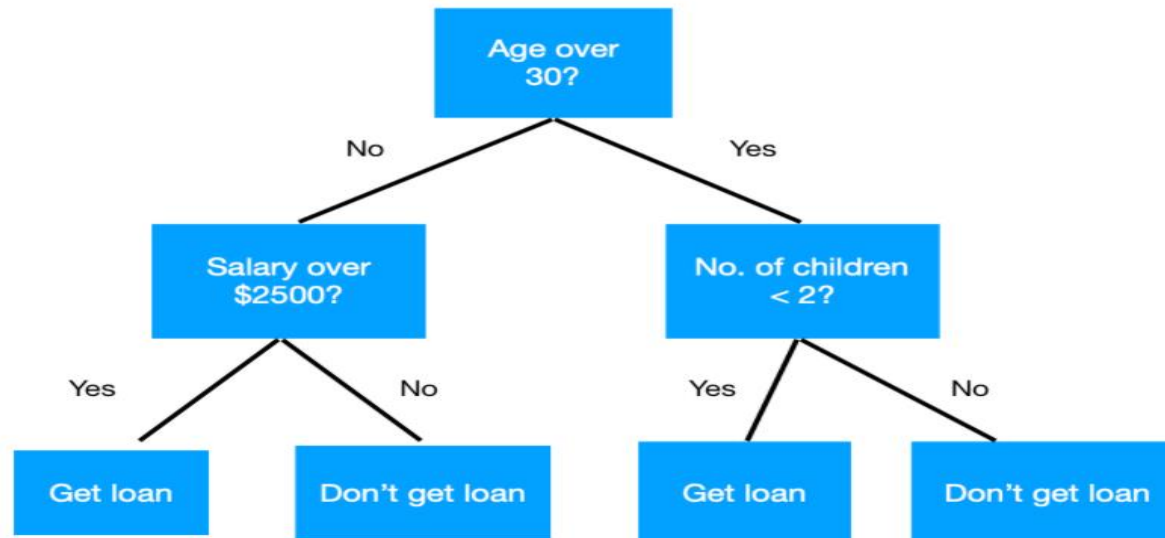
- Modèle assez robuste en grande dimension.
- Modèle nativement régularisé.
- Modèle paramétrable avec une variété de noyaux pour les cas non linéaires.

Inconvénients :

- L'entraînement est relativement coûteux en termes de temps.
- Très peu robustes pour les données déséquilibrées.

Arbres de décision

Les arbres de décisions essaient de séparer les données en se basant sur un ensemble de règles inférées.



Arbres de décision

Avantages :

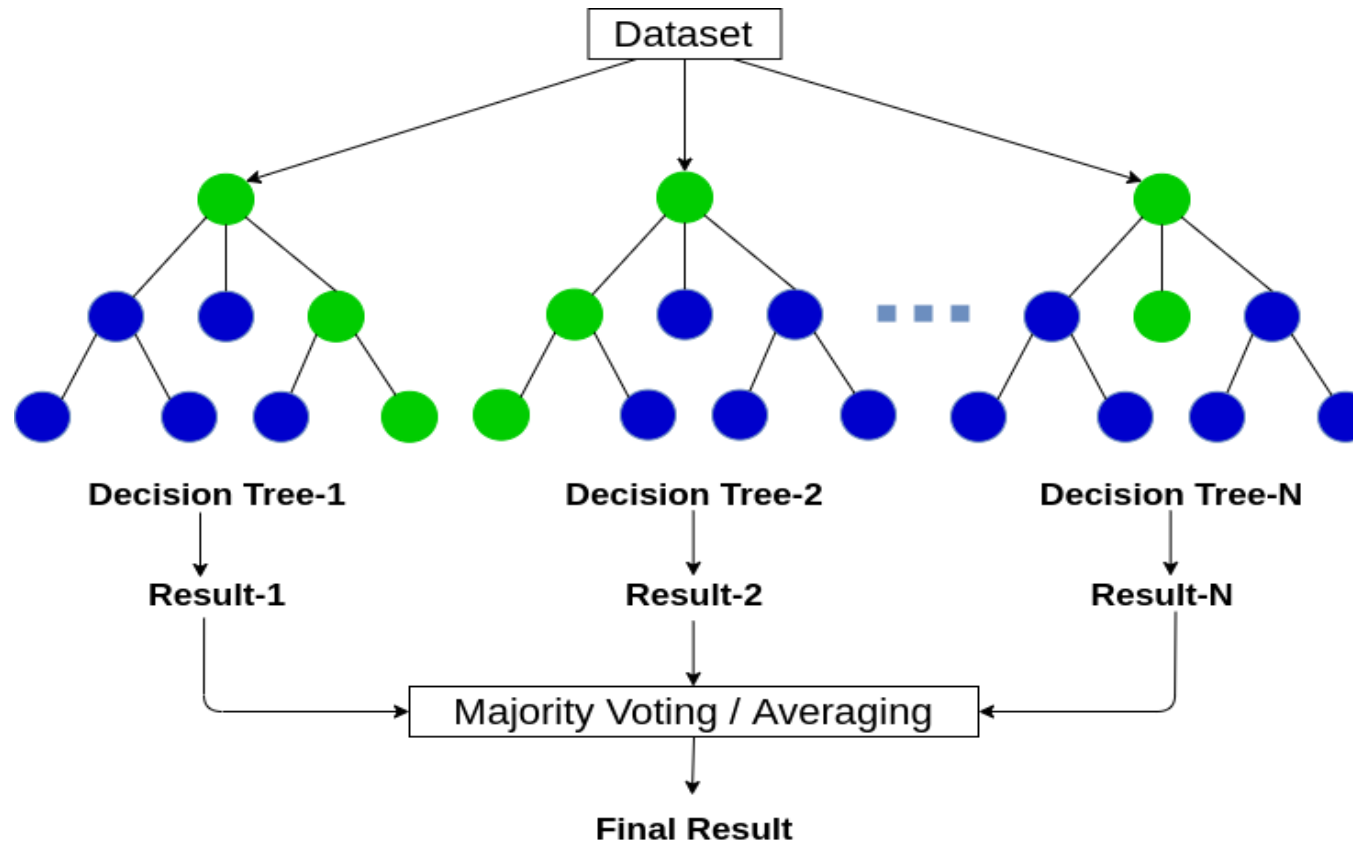
- Modèle simple et explicable.
- Modèle flexible et paramétrable.

Inconvénients :

- Modèle trop simple si les règles sous-jacentes sont complexes.
- Risque de surapprentissage relativement élevé.

Random forest

Random forest est une technique ensembliste qui fait appel à plusieurs arbre de décisions simultanément. Le résultat final est obtenu à l'aide d'un « vote » de tous les arbres.



Random forest

Avantages :

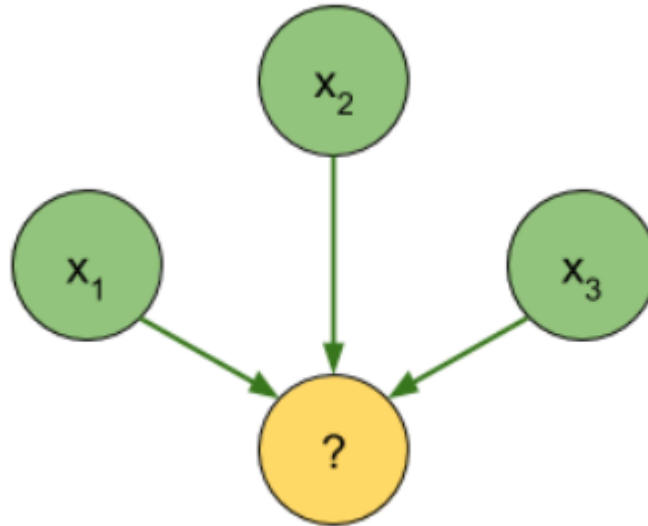
- Modèle très efficace pour les données tabulaires.
- Modèle flexible et paramétrable.
- Robustesse et variance relativement limitée.

Inconvénients :

- Modèle peu explicable.
- Modèle peu adapté aux données en grande dimension.

Naïve Bayes

Naïve Bayes (ou bayésien naïf) est un classifieur qui tente de séparer les données d'une manière linéaire en faisant l'hypothèse, très forte, que les caractéristiques (features) sont indépendantes.



Naïve Bayes

Avantages :

- Modèle simple à implémenter.
- Modèle très efficace notamment pour les données textuelles.

Inconvénients :

- L'hypothèse d'indépendance est rarement vérifiée.
- Très peu efficaces pour les données non linéaires.

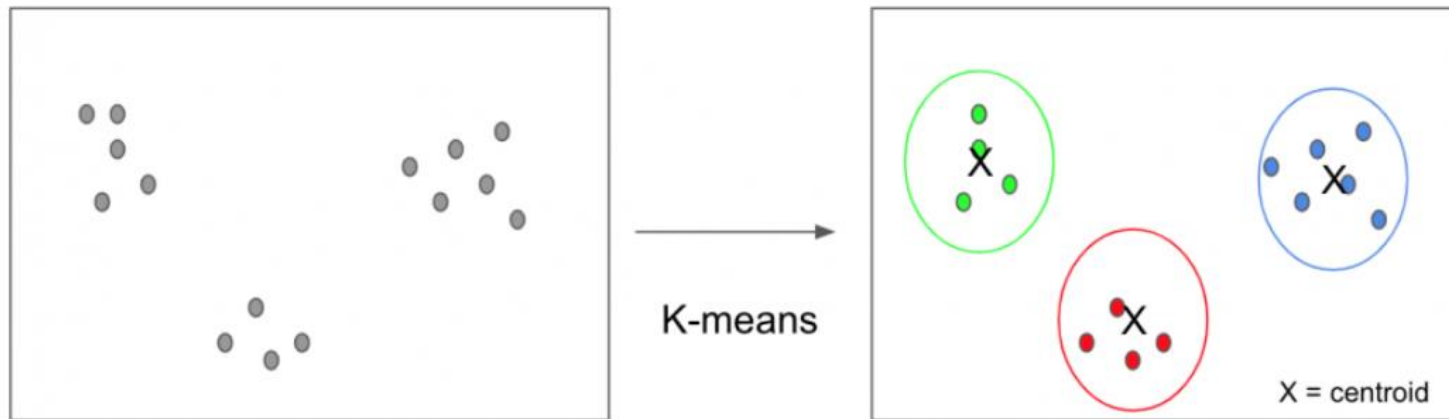
L'apprentissage non supervisé

L'apprentissage non supervisé consiste à laisser le modèle apprendre des patterns et régularités sous-jacentes aux données sans supervision humaine.

- Inputs : un jeu de données **non annotées**.
Exemple : des documents, des caractéristiques clients, etc.
- Output : des clusters de points, des outliers, etc.
Exemple : segments de clients, catégories de documents, etc.

K-means

K-means est un algorithme qui tente de regrouper les données en clusters en minimisant les distances entre les points d'un même cluster tout en maximisant les distances entre points appartenant à différents clusters.



K-means

Avantages :

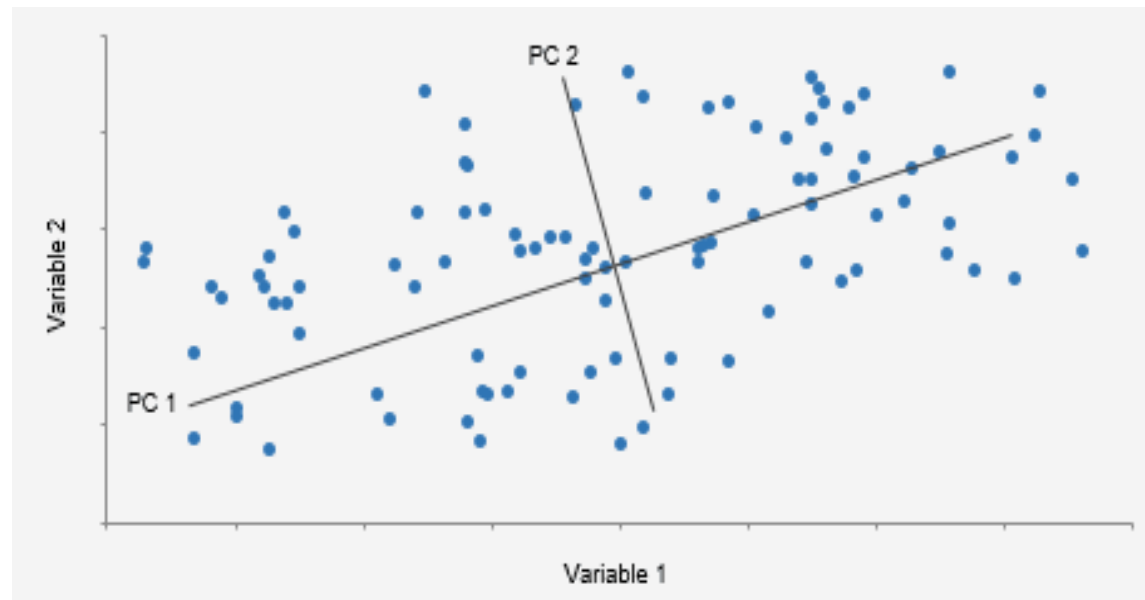
- Un algorithme efficace et simple à mettre en œuvre.
- Il converge toujours.

Inconvénients :

- Il n'est pas stable du fait de l'initialisation aléatoire (voir k-means++).
- Il n'est pas adapté aux données en grande dimension.

Analyse par composantes principales (ACP)

L'analyse par composantes principales consiste à projeter des données en grande dimension sur un espace de plus petite dimension, généralement deux ou trois, tout en gardant un maximum d'information (de variance). Elle est notamment utilisée à des fins de visualisation des données.



Analyse par composantes principaux (ACP)

Avantages :

- Un algorithme simple à mettre en œuvre.
- Très adapté aux données en grande dimension.

Inconvénients :

- Les composantes principales sont relativement difficiles à interpréter.
- Peu adaptée aux données non linéaires (voir ACP à noyau).

Représentations distribuées

One-Hot encoding

Dans les approches traditionnelles du NLP, les mots sont représentés par des vecteurs binaires dont les composantes sont 0 ou 1.

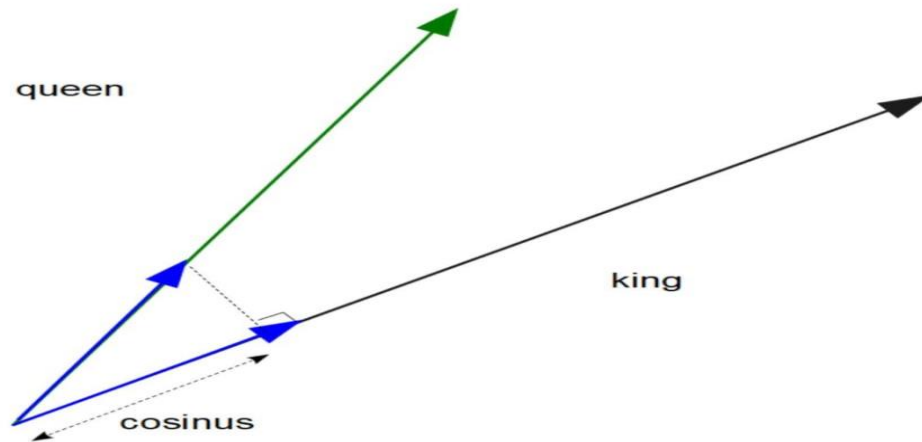
Par exemple, pour un vocabulaire de dimension $\sim 100\,000$

- Véhicule = $[0, 0, 1, 0, 0, 0, 0, 0, 0, \dots, 0, 0]$
- Voiture = $[0, 0, 0, 0, 0, 1, 0, 0, 0, \dots, 0, 0]$

Cette representation ne permet pas de prendre en compte la dimension sémantique.

Similarités sémantiques

Nous sommes intéressés par des représentations de mots qui capturent la distance sémantique, sous forme de produit scalaire par exemple (ou distance cosine).



Représentations distribuées (word embeddings)

*« You shall know a word by the company it keeps »
J.R. Firth (1957)*

Les mots sont similaires s'ils apparaissent fréquemment dans le même contexte.

- Il conduit son **véhicule** pour rentrer à la maison.
- Il conduit sa **voiture** pour rentrer chez lui.

Construction d'une représentation distribuée

Considérons ce corpus à titre d'exemple :

cnn in crop analysis

cnn and svm are widely used

linear_regression performed along with svm

linear regression for crop and farm

svm being used for farm monitoring

Do cnn, svm and linear_regression appear in the same context?

Le vocabulaire est alors :

[cnn, in, crop, analysis, and, svm, are, widely, used, linear_regression, performed, along, with, for, farm, being, monitoring, do, appear, the, same, context]

dim(vocabulaire) = 22

Matrice de co-occurrences

La matrice de co-occurrence représente la fréquence à laquelle les mots apparaissent ensemble deux à deux.

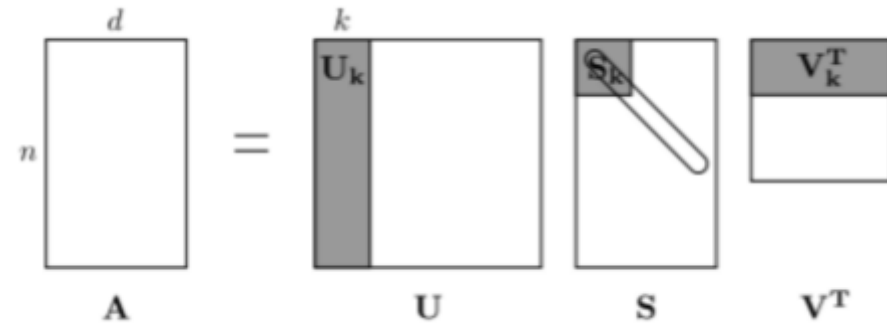
```
      cnn in crop ...
cnn  [[0, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1],
in    [2, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1],
crop  [1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0],
...    [1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],
      [2, 1, 0, 0, 1, 0, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
      [1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [1, 0, 0, 0, 1, 2, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0],
      [1, 1, 1, 0, 1, 2, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1],
      [0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 2, 1, 1, 0, 0, 0, 0],
      [0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 2, 0, 1, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0],
      [1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1],
      [1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1],
      [1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1],
      [1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0]]
```

Réduction de la dimension

La décomposition en valeurs singulières est une technique permettant de réduire la dimension des données (similaire à l'ACP).

Etant donné une matrice $\mathbf{A} \in \mathbb{R}^{n \times d}$

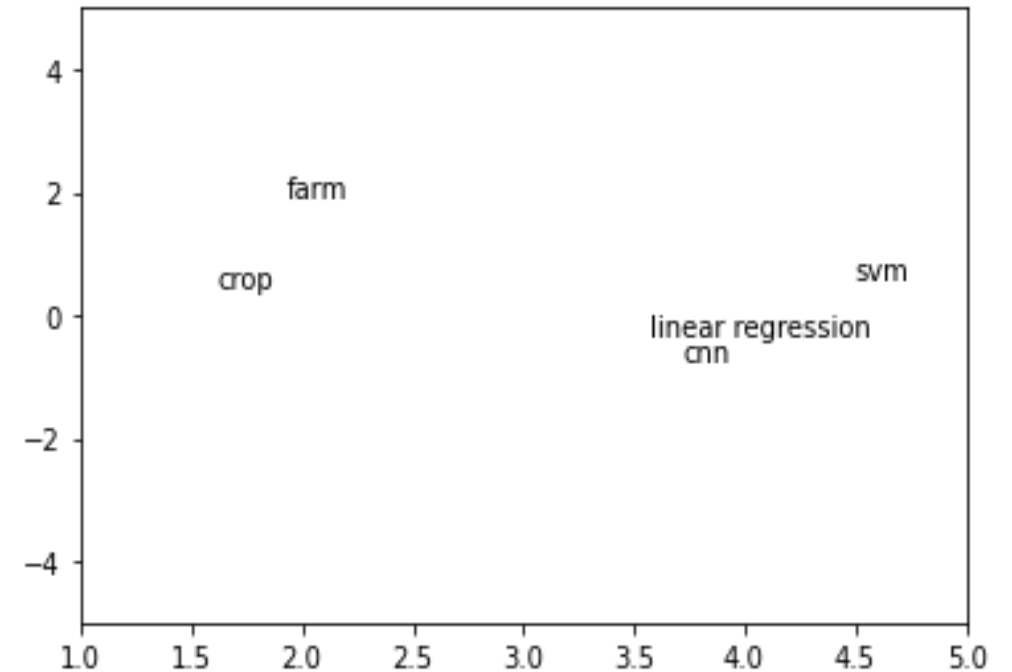
$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T \quad \text{où} \quad \mathbf{U} \in \mathbb{R}^{n \times r}, \mathbf{D} \in \mathbb{R}^{r \times r}, \mathbf{V} \in \mathbb{R}^{d \times r}.$$



\mathbf{U} est la matrice contenant les représentations (vecteurs de mots)

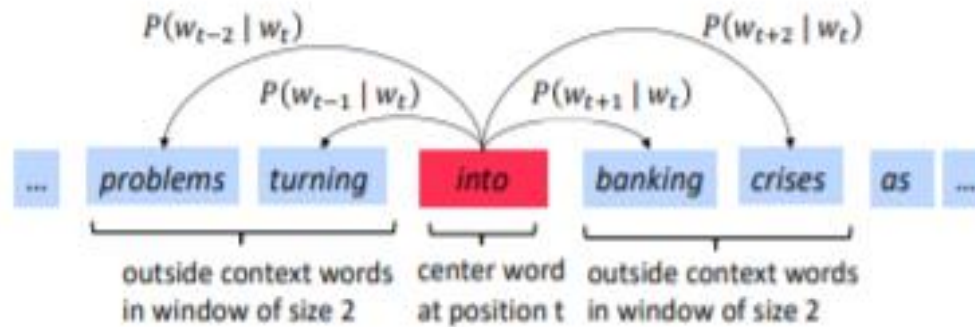
Visualisation des représentations distribuées

```
cnr [ 3.72113518, -0.73233585],  
ln [ 2.7940387 , -1.40864784],  
crop [ 1.61178082, 0.44786021],  
... [ 0.77784994, -0.31230389],  
[ 2.12245048, 1.29571556],  
[ 4.49293777, 0.58090417],  
[ 1.33312748, 0.66758743],  
[ 1.33312748, 0.66758743],  
[ 2.25882809, 1.80738544],  
[ 3.56593605, -0.31006976],  
[ 0.95394179, 0.079159 ],  
[ 0.95394179, 0.079159 ],  
[ 0.95394179, 0.079159 ],  
[ 1.92921553, 1.94783497],  
[ 1.92921553, 1.94783497],  
[ 1.12301312, 1.42126096],  
[ 1.12301312, 1.42126096],  
[ 2.26025282, -1.31571175],  
[ 2.26025282, -1.31571175],  
[ 2.26025282, -1.31571175],  
[ 2.26025282, -1.31571175],  
[ 2.26025282, -1.31571175]
```



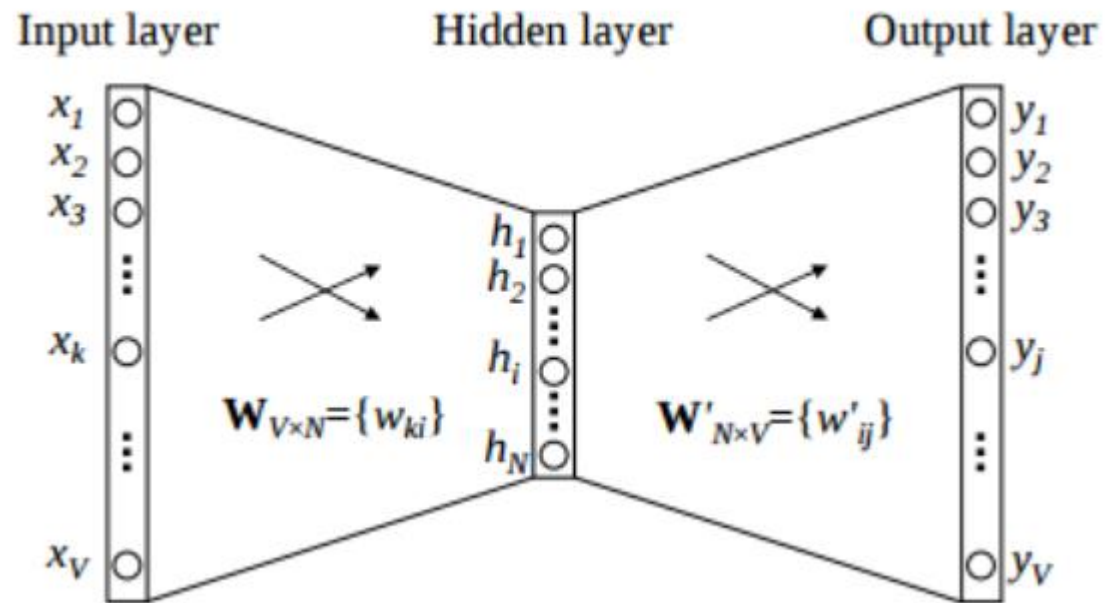
Skip-gram et Continuous Bag of Words (CBOW)

Prédire un mot masqué dans une phrase sachant le contexte (les autres mots).



Continuous Bag of Words (CBOW)

Calcul des représentations à l'aide des réseaux de neurones.



Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov

Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen

Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado

Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean

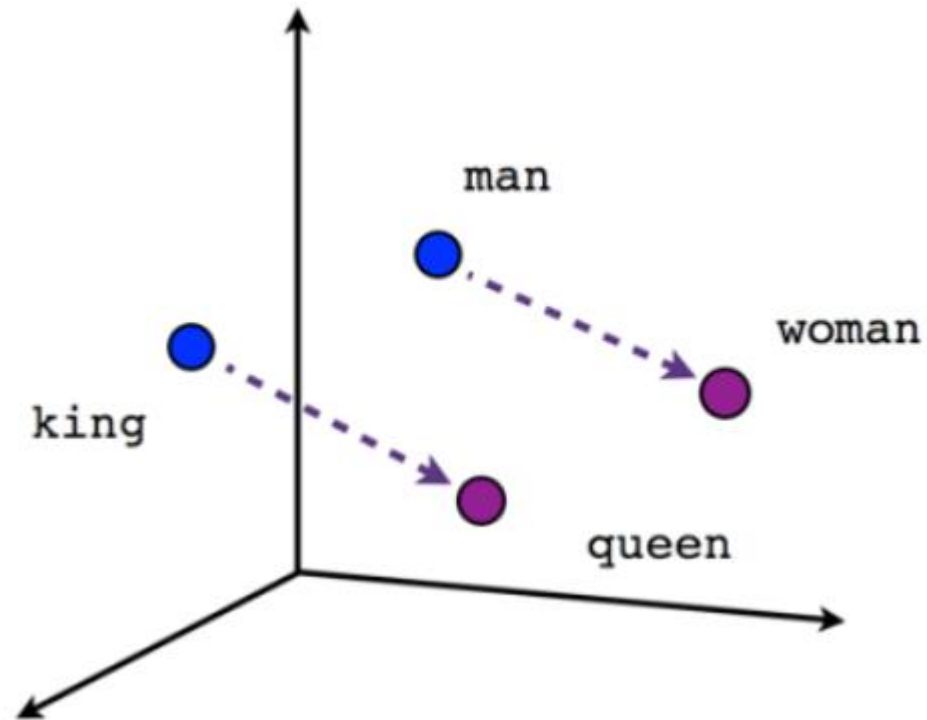
Google Inc., Mountain View, CA
jeff@google.com

Abstract

We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.

Word2Vec

Le modèle Word2Vec permet de prendre en compte la dimension sémantique entre les mots.



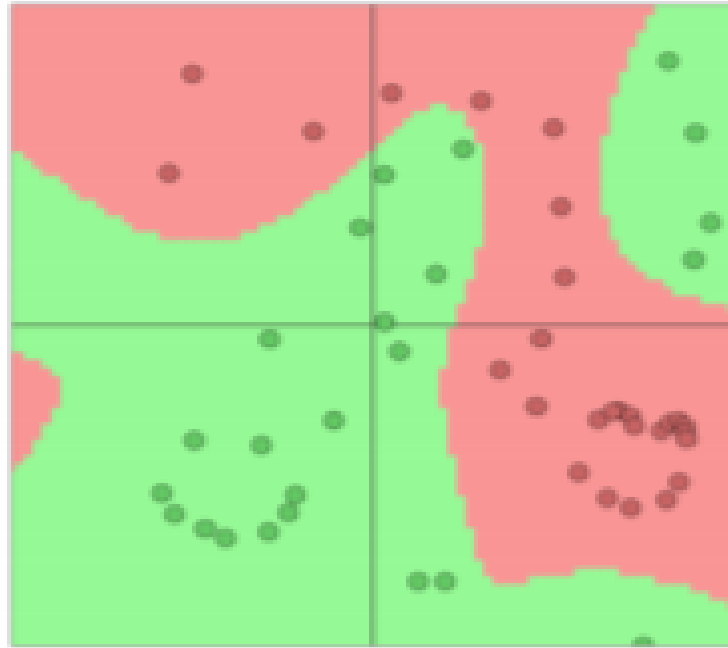
Autres représentations

- Glove (2014) : proposé par une équipe de Stanford, il combine à la fois les techniques modernes de deep learning et les techniques statistiques (co-occurrences entre les mots). Il permet d'avoir des représentations des mots plus globales que Word2Vec.
- Fasttext (à partir de 2016) : la librairie a été créée par Facebook. Le modèle est entraîné sur des subwords (n-grams de caractères). Il est ainsi plus efficace pour traiter les mots inconnus (out of vocabulary).
- Représentations contextuelles (Transformers), etc.

NLP par le deep learning

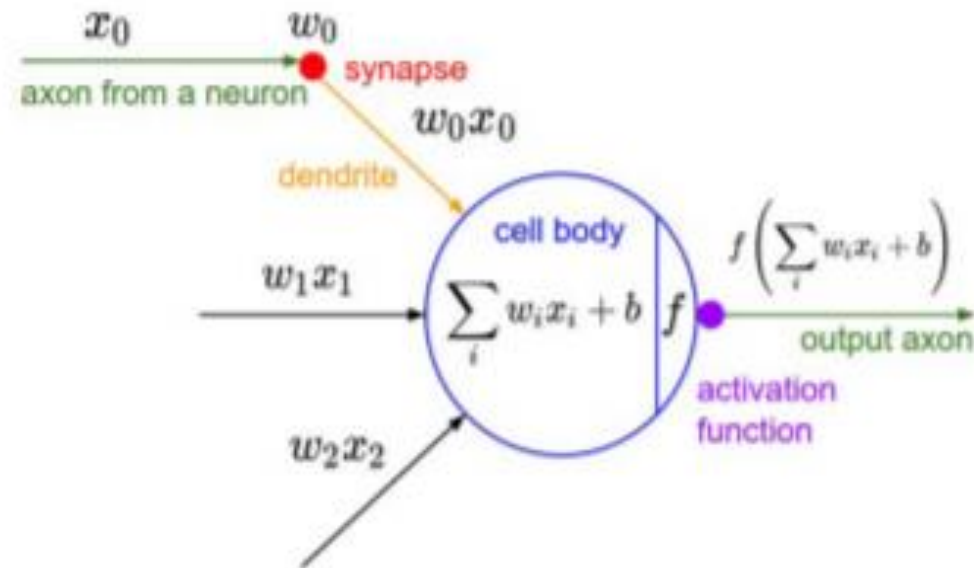
Éléments sur les réseaux de neurones

Les réseaux de neurones sont des modèles capables de représenter des patterns très complexes. Ils sont particulièrement adaptés aux données non structurées et en grande dimension.



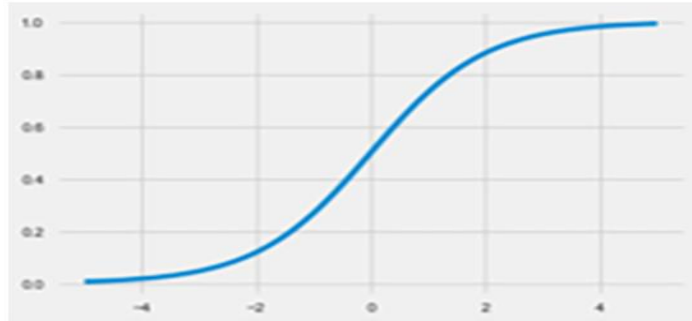
Concept des réseaux de neurones

Les réseaux de neurones sont composées de cellules élémentaires, les neurones, permettant de réaliser des opérations simples (addition, multiplication).

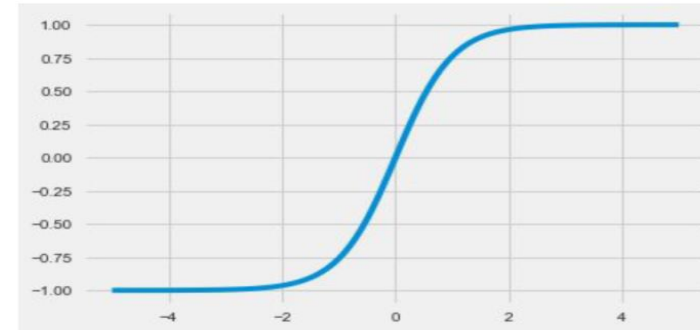


Fonctions d'activation

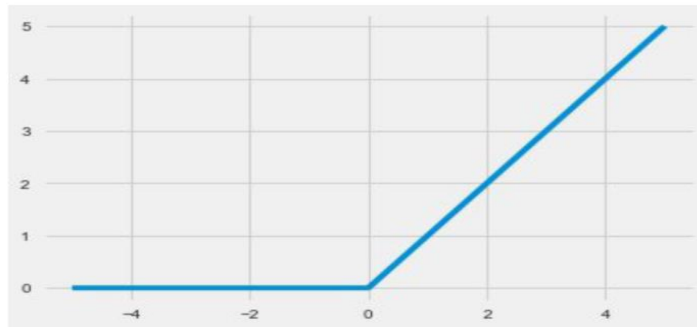
Sigmoïde



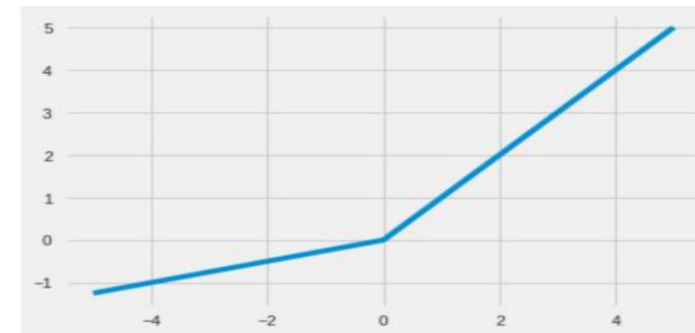
Tanh



ReLU



ReLU paramétrique

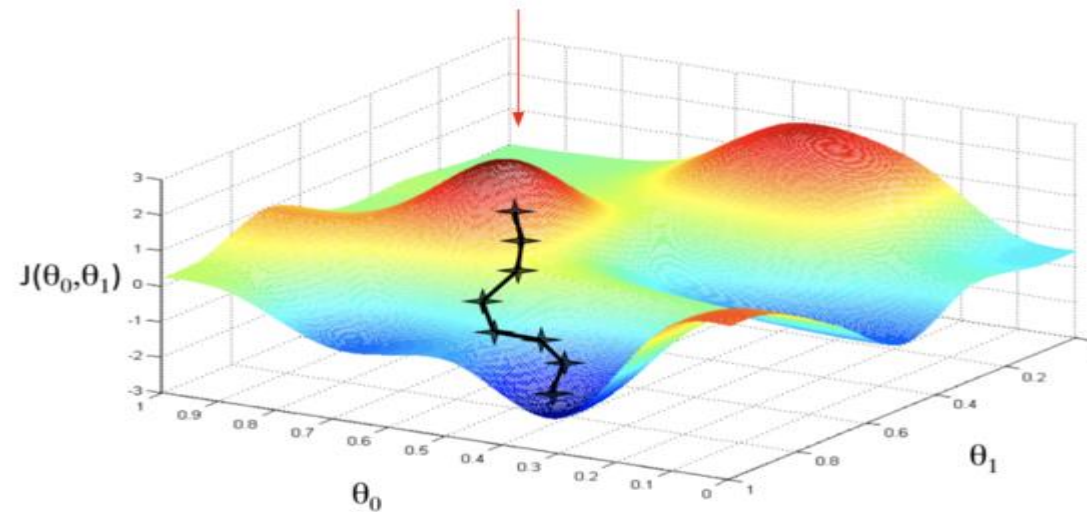


$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^k e^{x_j}}$$

Entraînement des réseaux de neurones : descente du gradient

$$\theta^{new} = \theta^{old} - \alpha \nabla_{\theta} J(\theta) \quad \nabla_{\theta} J(\theta) \text{ est calculé par backpropagation}$$

Learning rate



Backpropagation

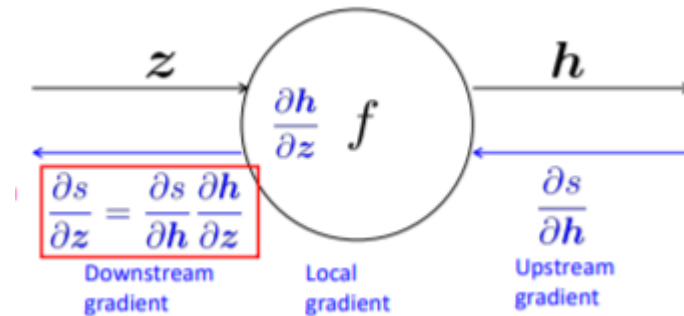
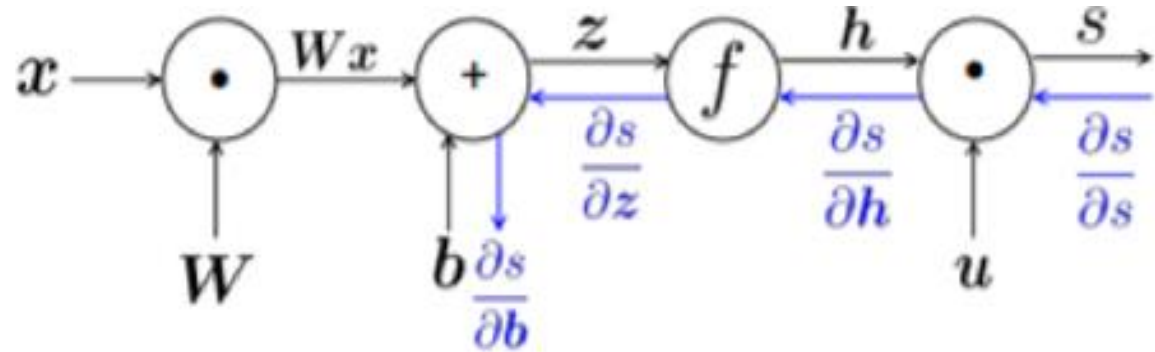
Les paramètres sont mis à jour (calculés) en partant de l'erreur et en remettant les différentes couches du réseau.

$$s = u^T h$$

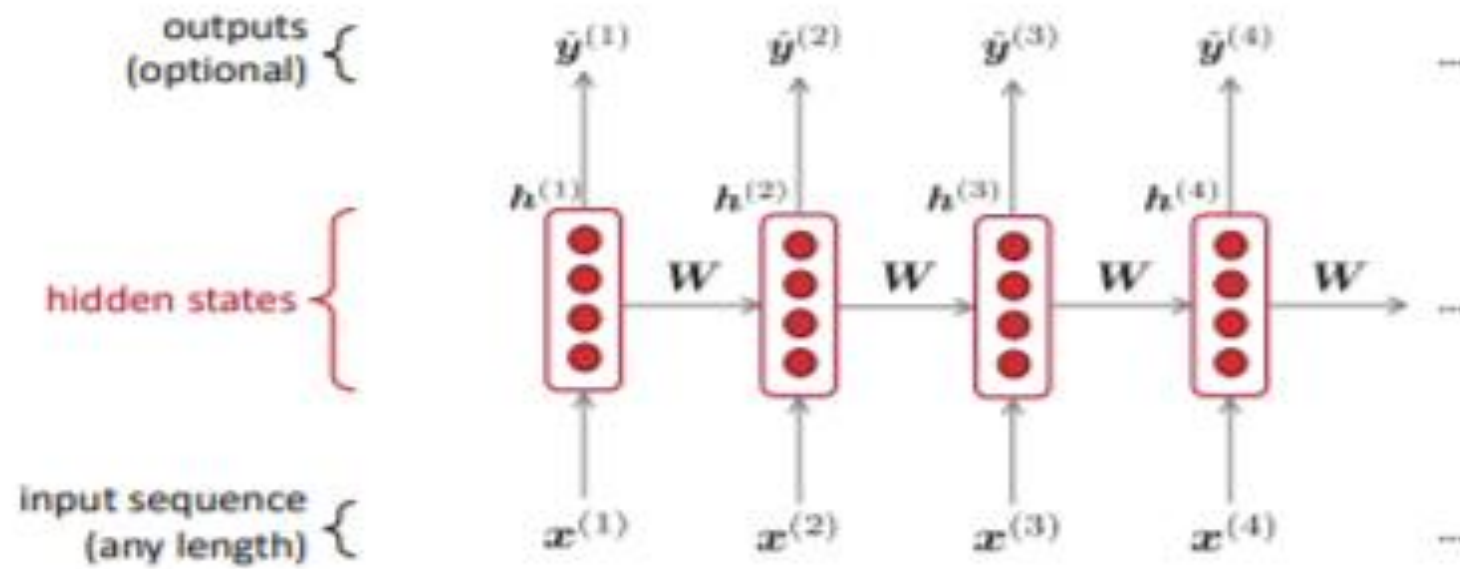
$$h = f(z)$$

$$z = Wx + b$$

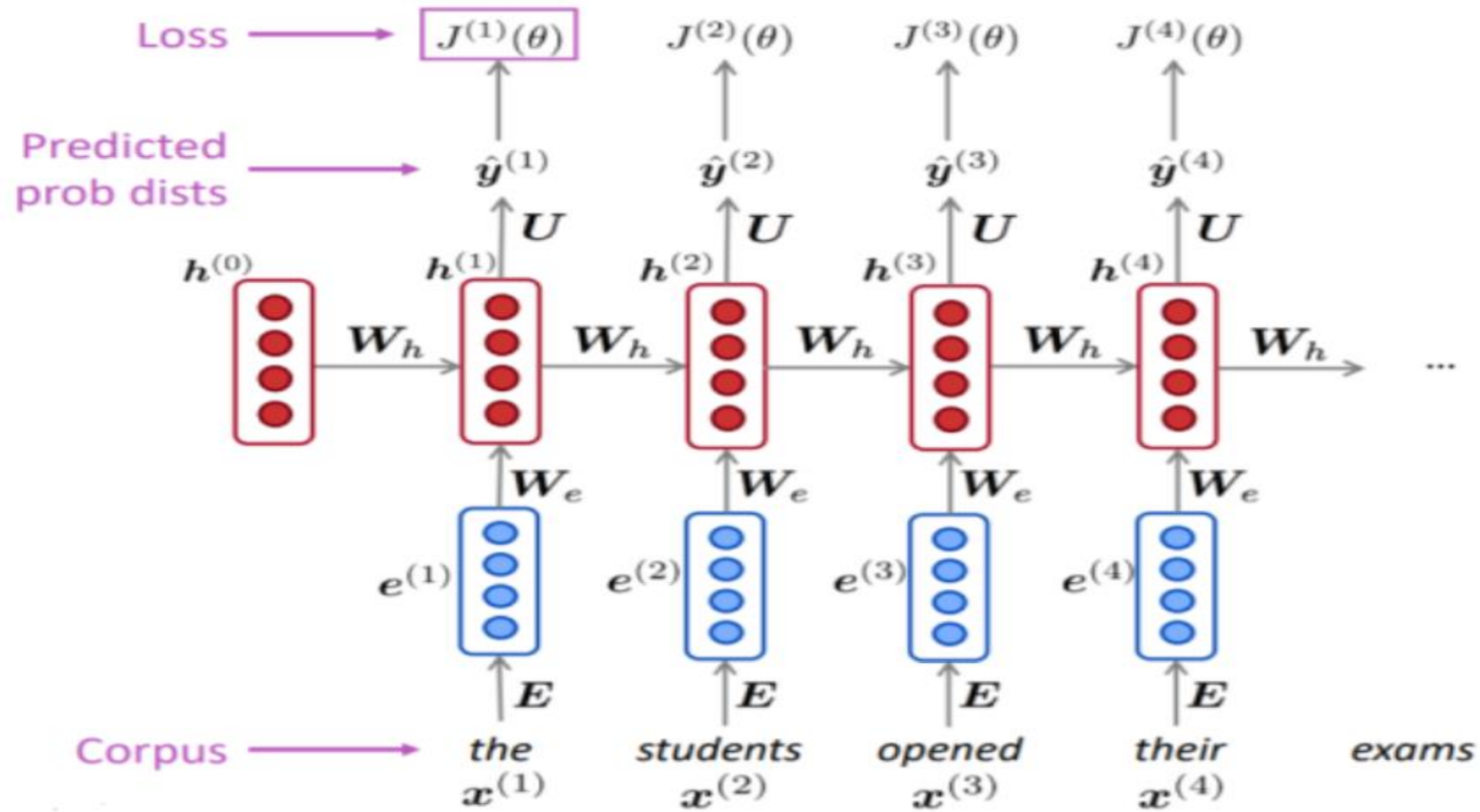
x (input)



Les réseaux de neurones récurrents (RNN)



Entraînement des RNNs



RNNs pour la génération de texte

Title: CHOCOLATE RANCH BARBECUE

Categories: Game, Casseroles, Cookies, Cookies

Yield: 6 Servings

2 tb Parmesan cheese -- chopped

1 c Coconut milk

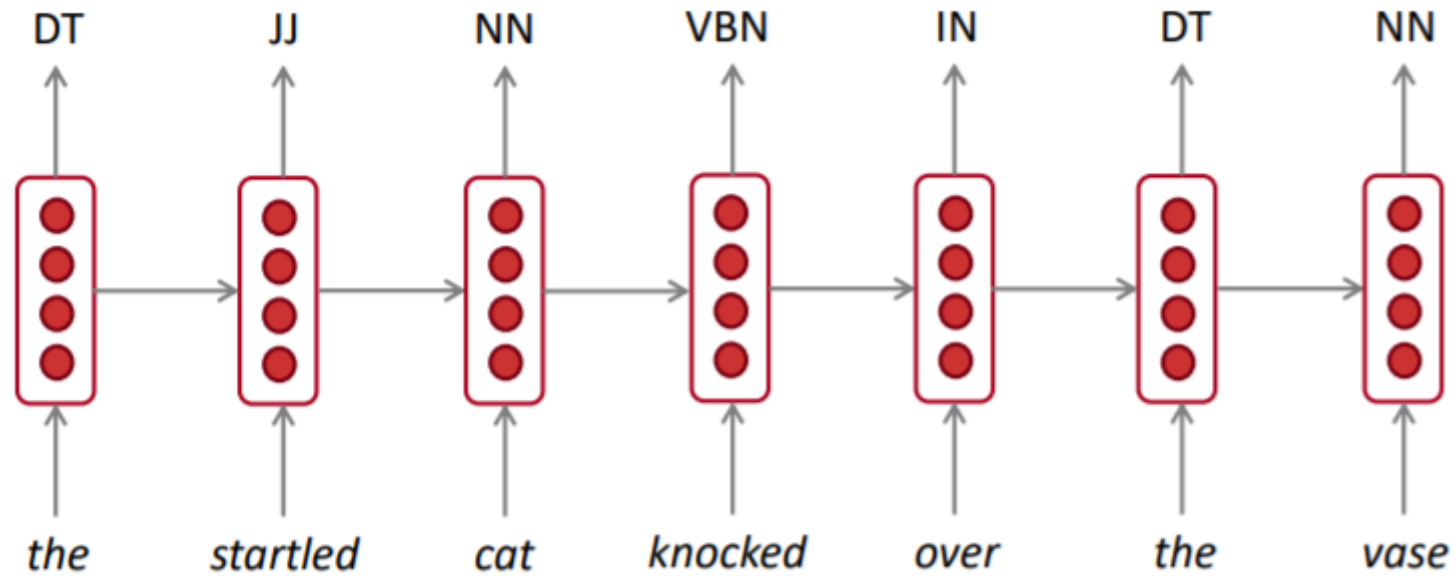
3 Eggs, beaten

Place each pasta over layers of lumps. Shape mixture into the moderate oven and simmer until firm. Serve hot in bodied fresh, mustard, orange and cheese.

Combine the cheese and salt together the dough in a large skillet; add the ingredients and stir in the chocolate and pepper.

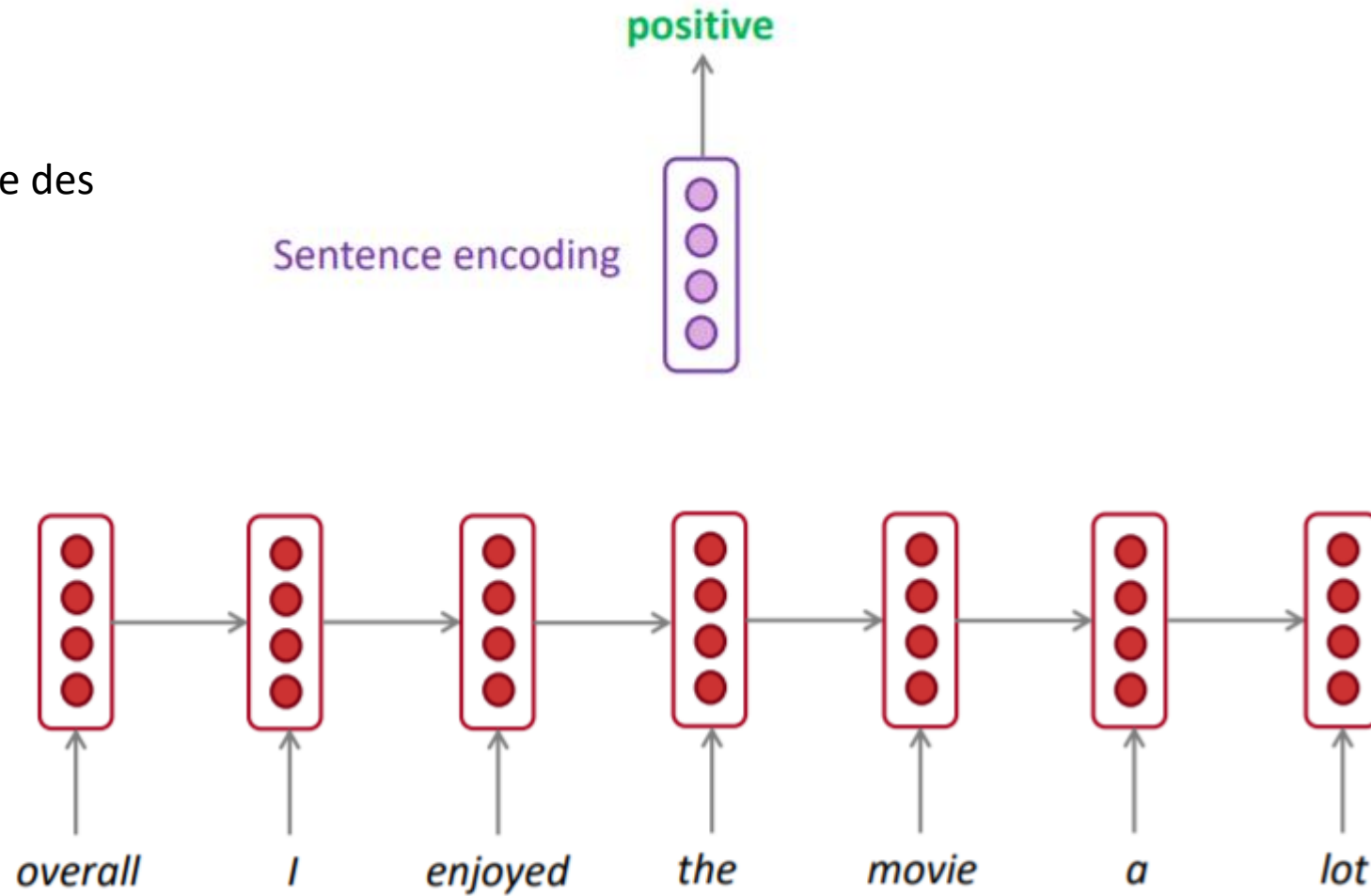
Source <https://gist.github.com/nylki/1efbaa36635956d35bcc>

RNNs pour le part of speech tagging



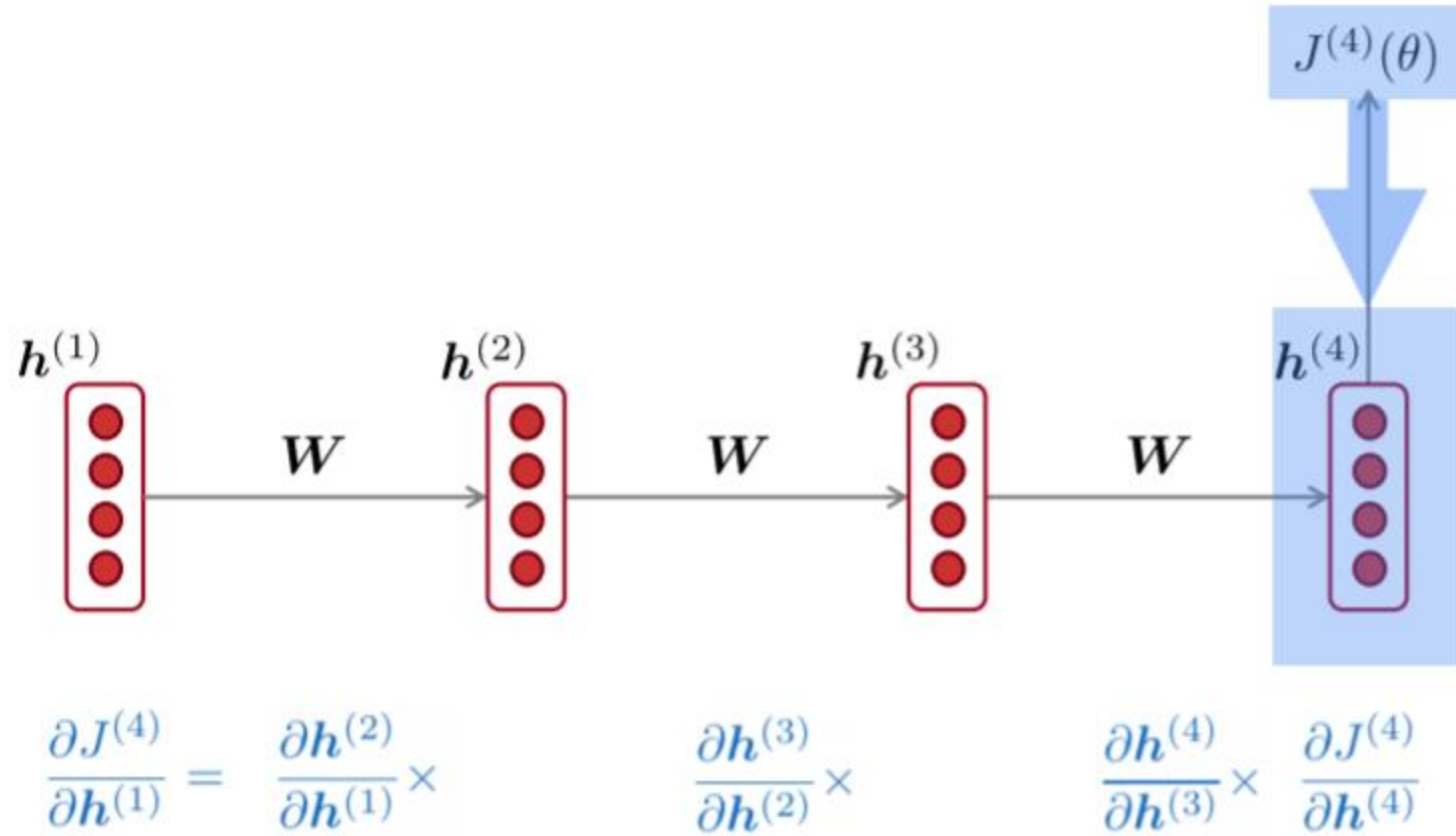
RNNs pour la classification de texte

Exemple : analyse des sentiments



Limitations des RNNs

La principale limitation des RNNs est l'annulation et l'explosion du gradient.



Long Short-Term Memory (LSTM)

Les LSTMs sont une variante de RNNs, introduits en 1997 pour remédier au problème de l'annulation/explosion du gradient.

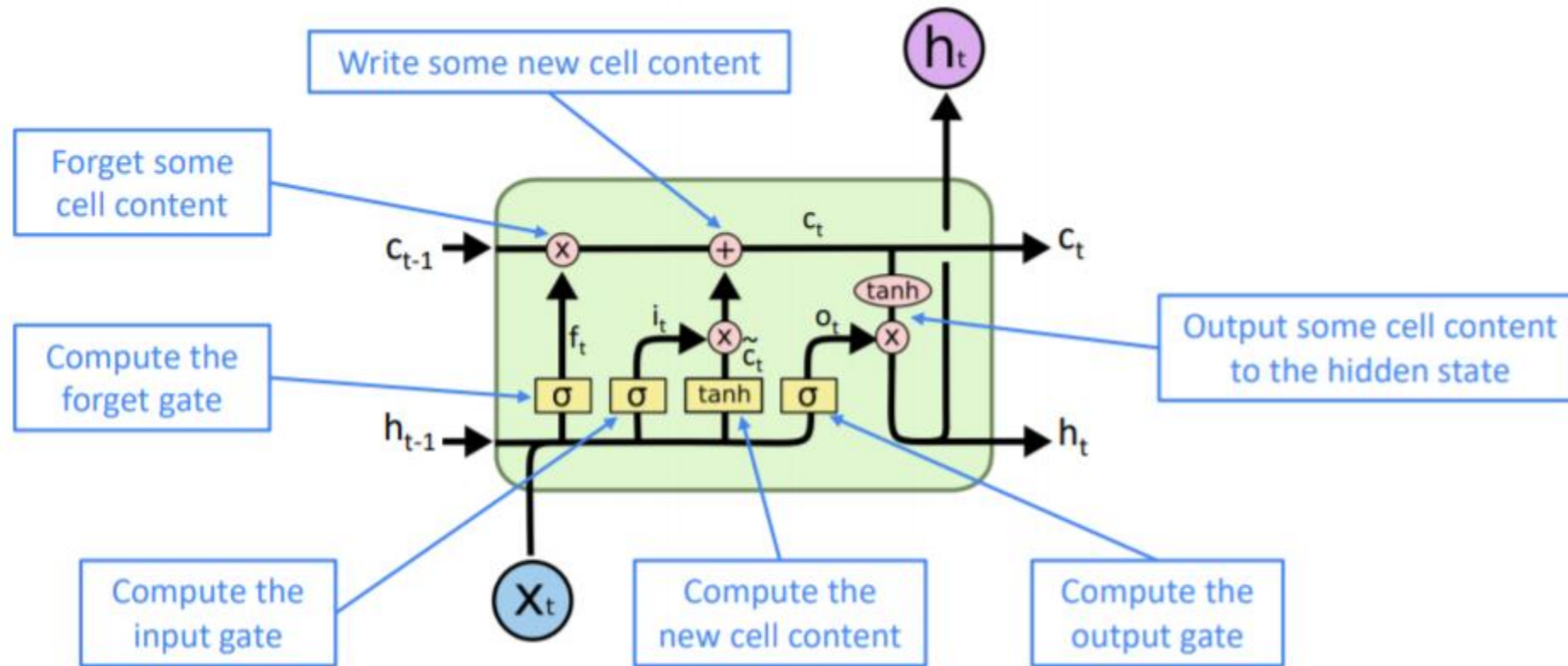
A chaque étape t , on dispose d'un hidden state d'un cell state:

- Tous les deux sont des vecteurs d'une certaine taille.
- Le cell state mémorise les informations de long terme.
- Le LSTM peut supprimer ou récupérer une information à partir du cell state.

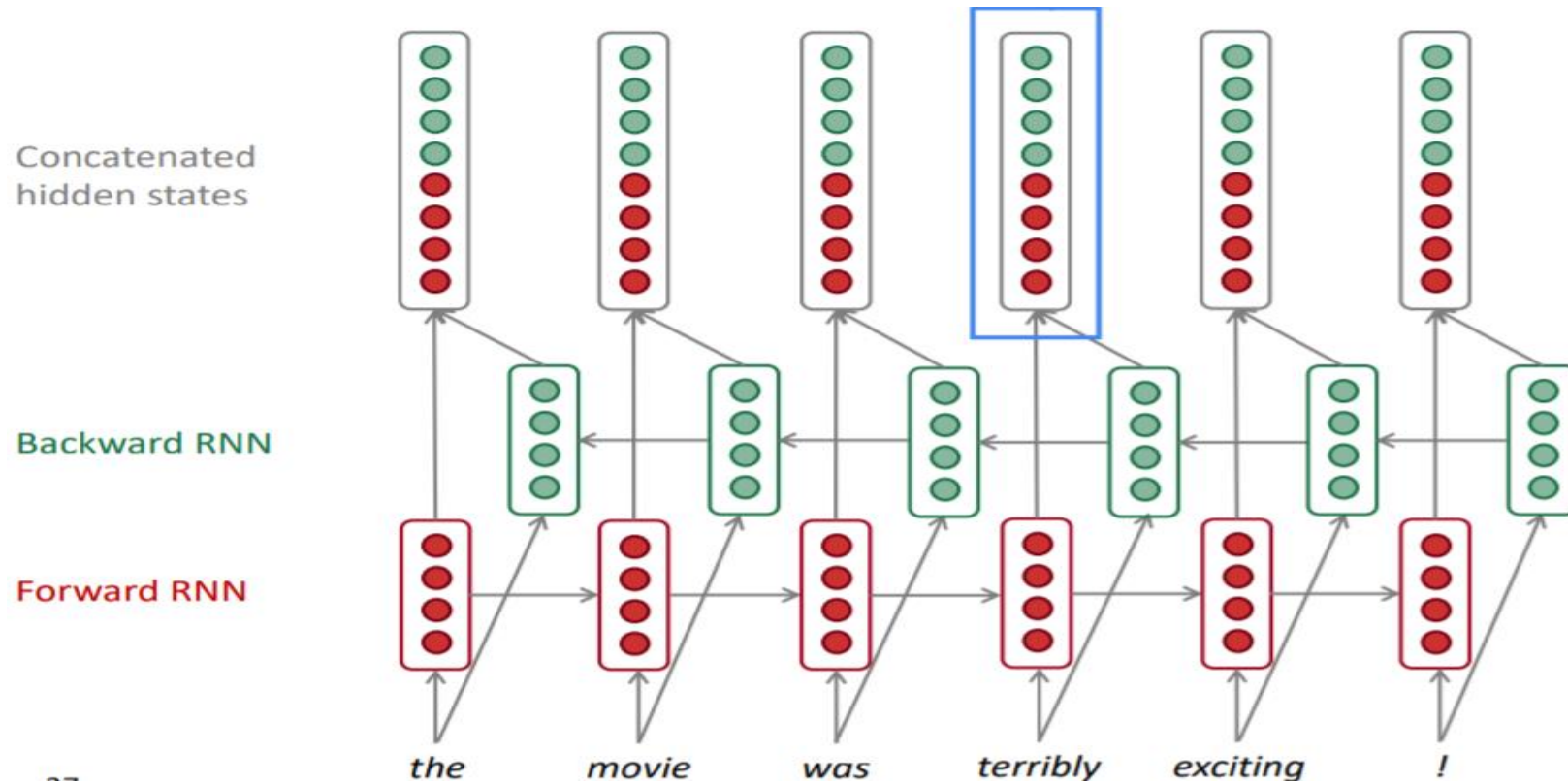
Ces opérations sont contrôlées par des portes logiques :

- Qui sont également des vecteurs.
- Pour chaque étape t , la porte peut être ouverte, fermée ou partiellement ouverte pour contrôler le flux d'information.

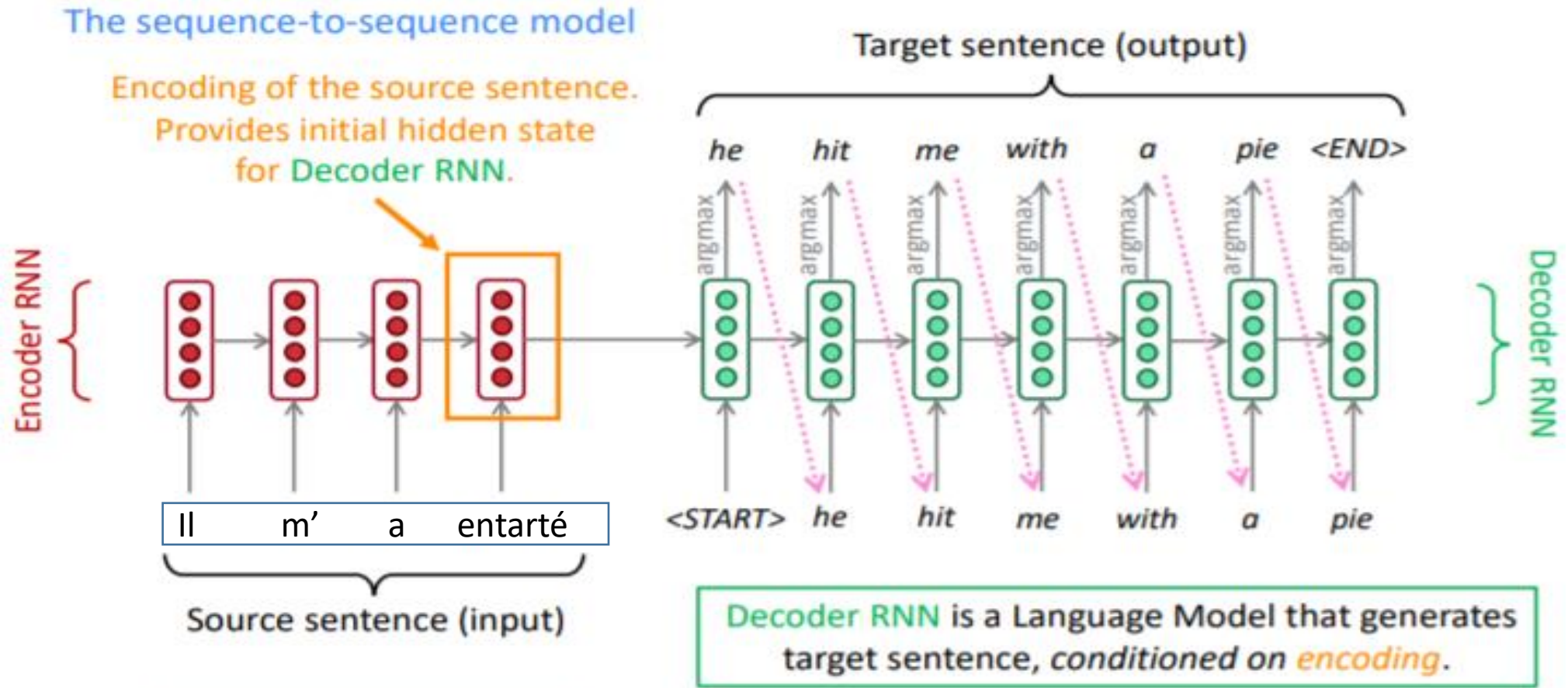
Long Short-Term Memory (LSTM)



Bidirectional Recurrent Neural Networks



Les modèles sequence-to-sequence models (seq2seq)

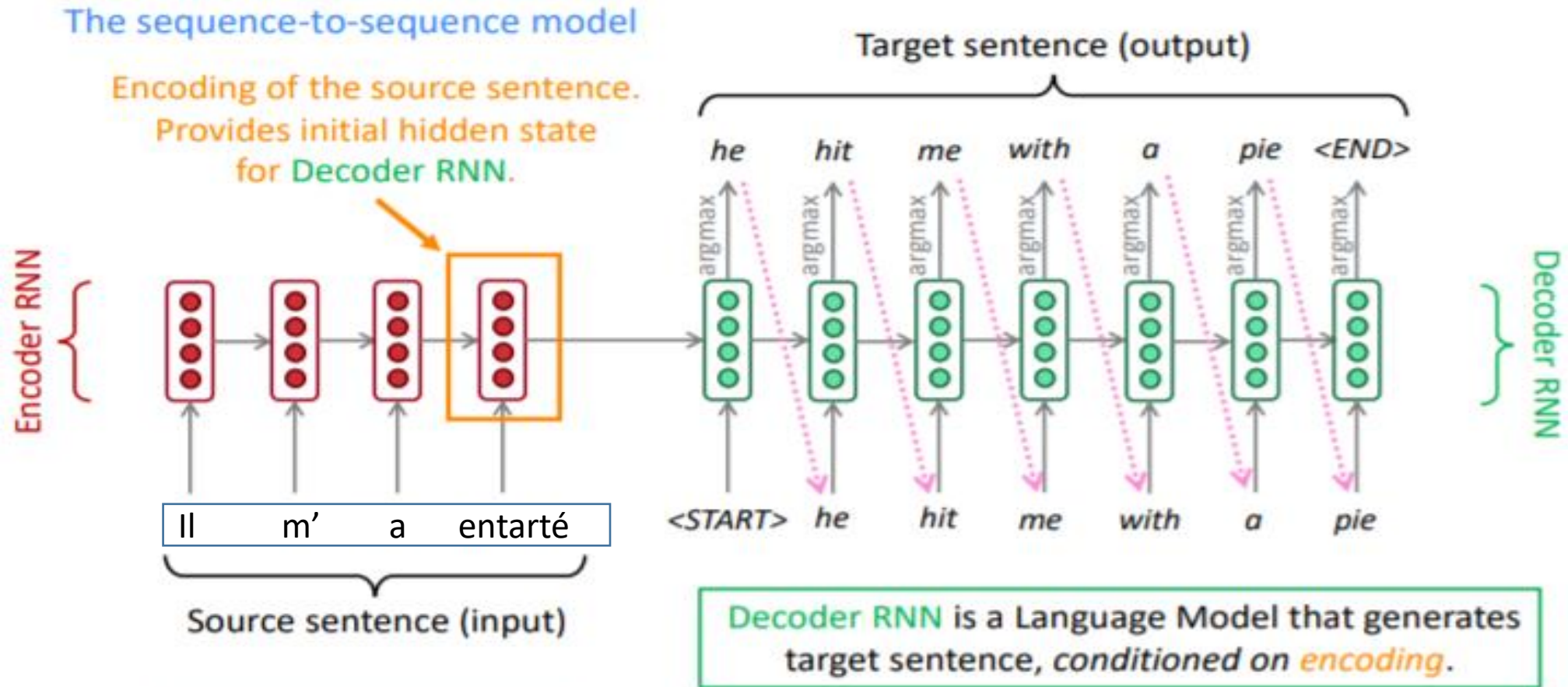


Les modèles sequence-to-sequence (seq2seq)

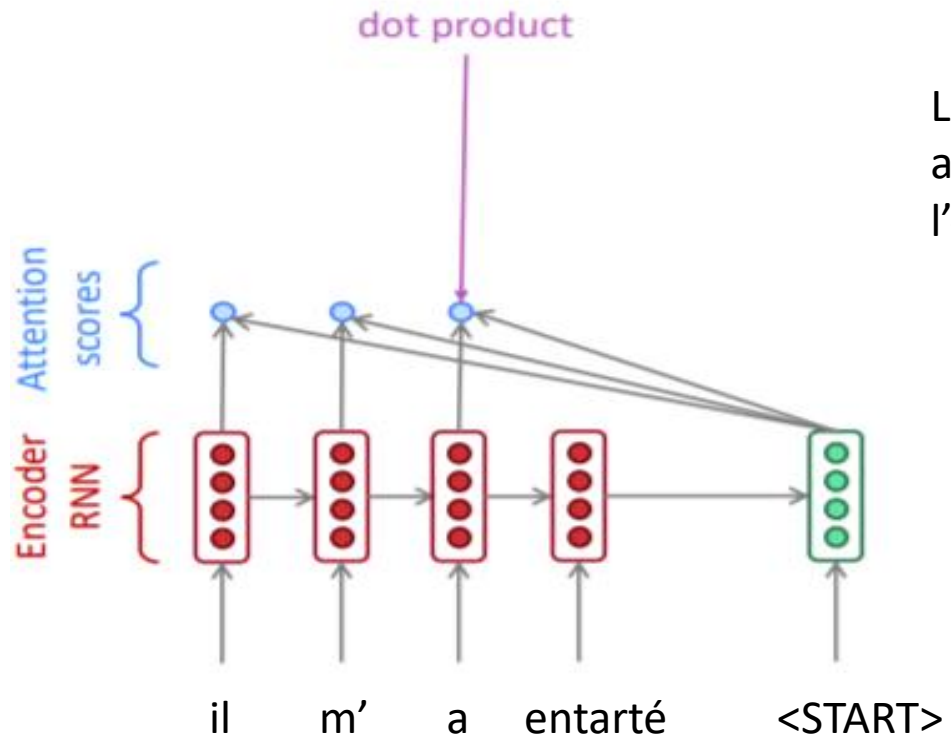
Les modèles seq2seq sont notamment adaptés pour :

- La traduction (Machine Translation): Traduire un texte de l'anglaise au français.
- Résumé de texte (Texte long en input et texte court en output).
- Question/réponse (question en input et réponse en output).
- Génération de texte (langage naturel en input et code Python en output).
- Etc.

Limitations des modèles sequence-to-sequence

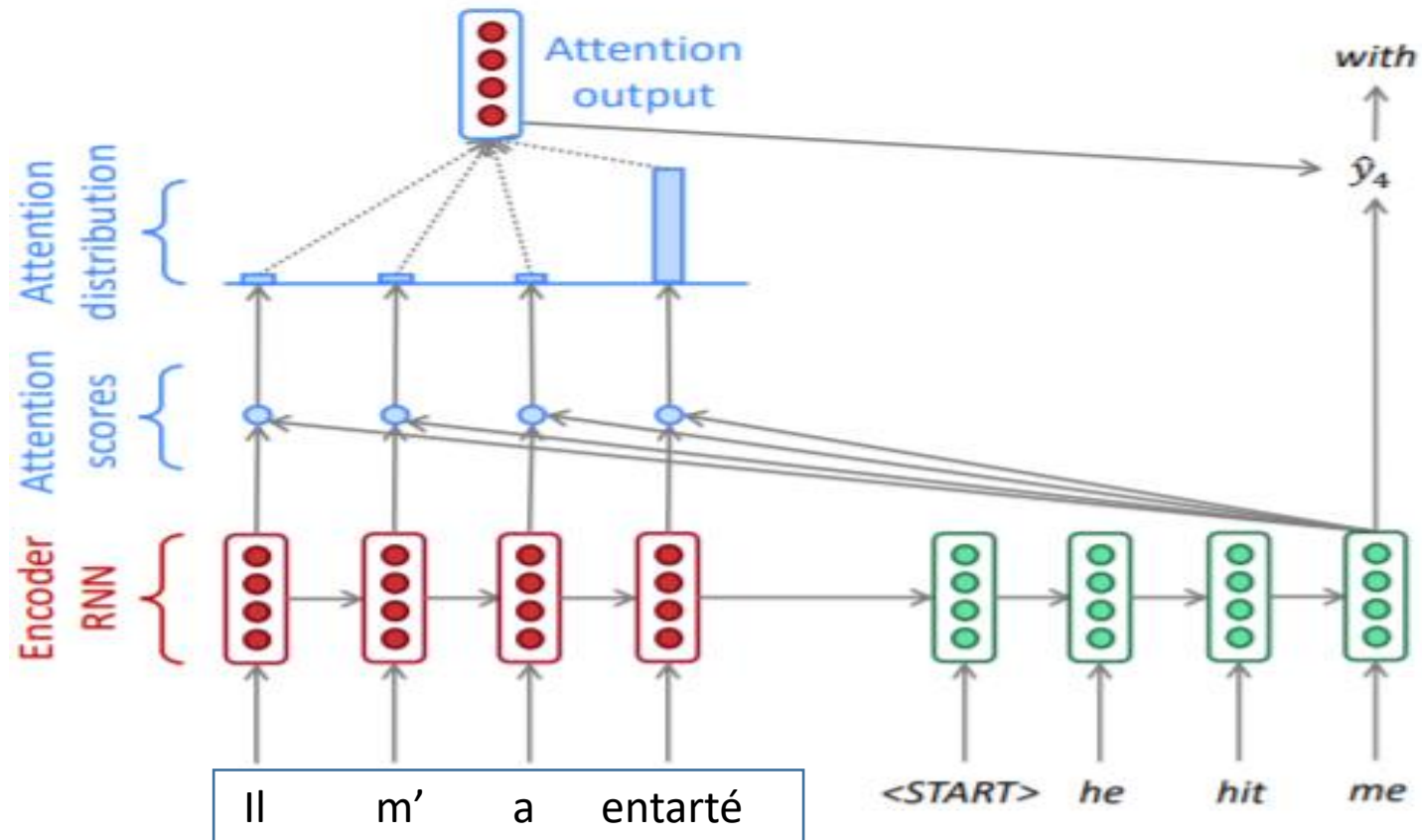


Le mécanisme d'attention



Le mécanisme d'attention offre la possibilité au décodeur d'accéder à plusieurs mots dans l'encodeur.

Le mécanisme d'attention



Apprentissage auto-supervisé et Transformers

Concept de l'apprentissage auto-supervisé

- Limites de l'apprentissage supervisé : le deep learning est très gourmand en données pour l'entraînement des modèles. L'annotation des données est rédhibitoire dans certains cas.
- L'apprentissage supervisé exploite « l'annotation naturelle » des données.

L'étudiant a [] son livre

- Le modèle est entraîné en prédisant des mots masqués aléatoires dans le corpus. Cet apprentissage est particulièrement adapté aux données textuelles.

Emergence des Transformers

Xiv:1706.03762v5 [cs.CL] 6 Dec 2017

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaizer@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Architecture du Transformer originel

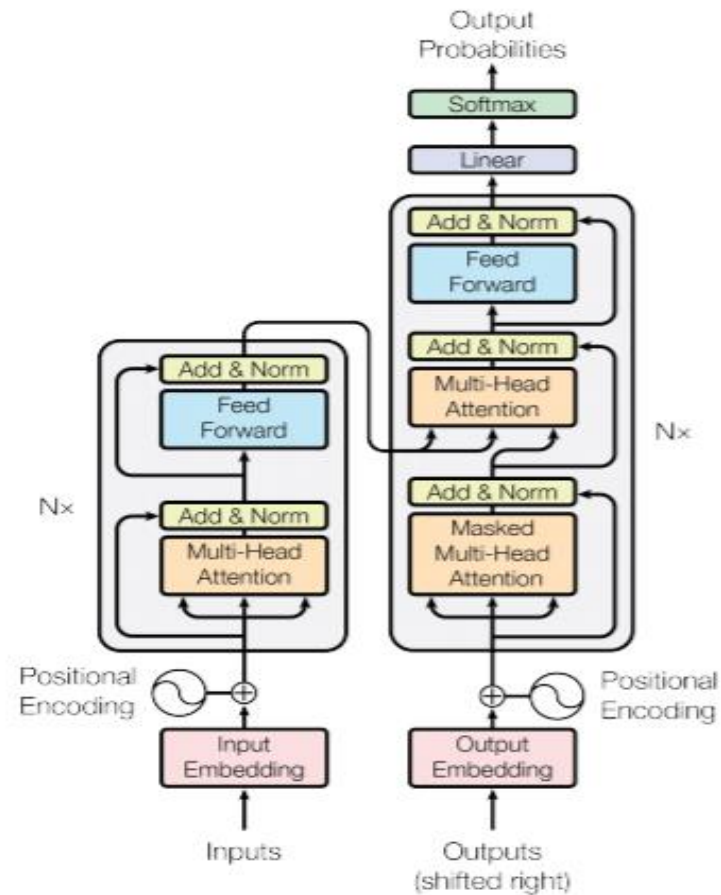
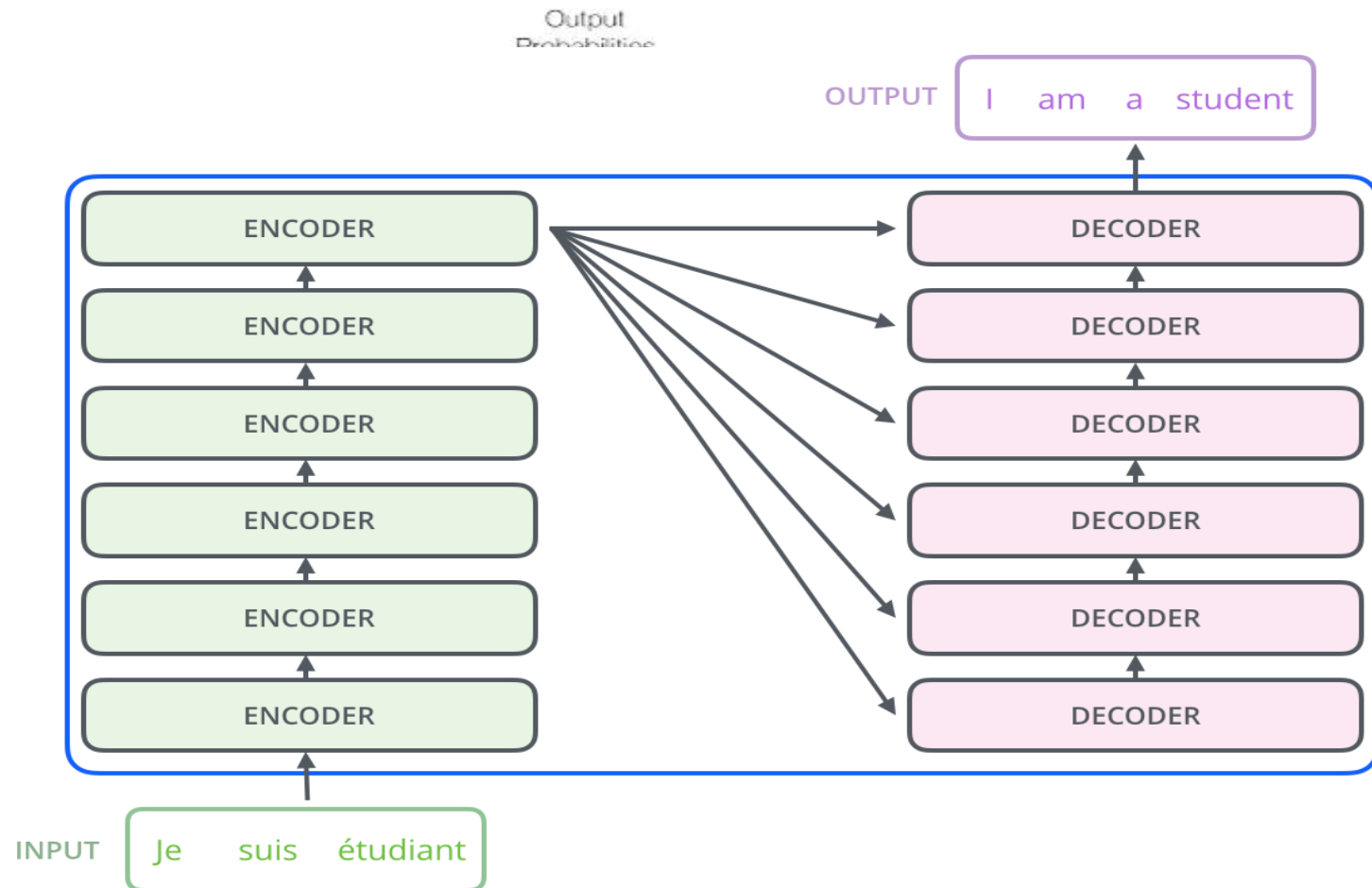


Figure 1: The Transformer - model architecture.

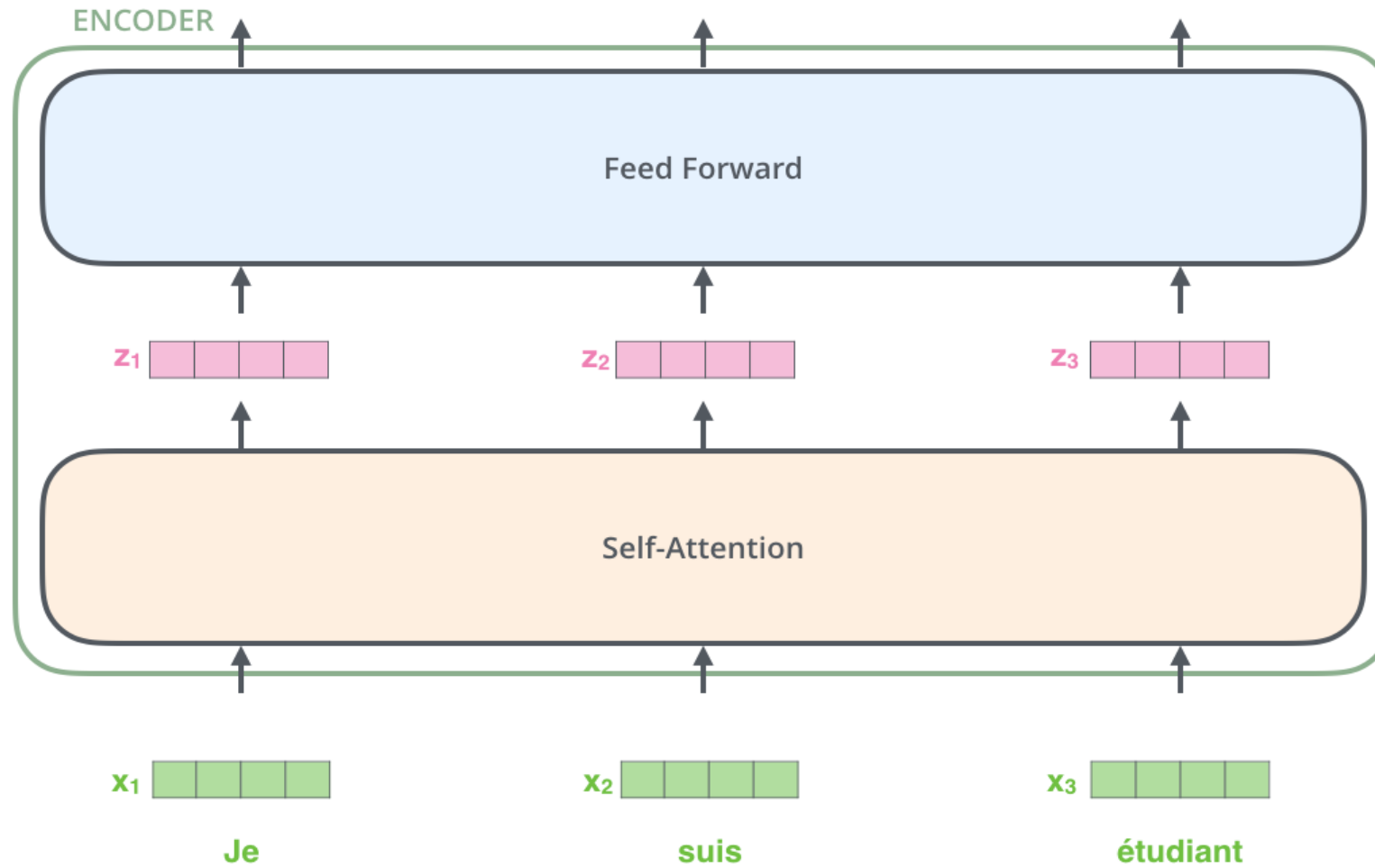
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Mécanisme d'attention

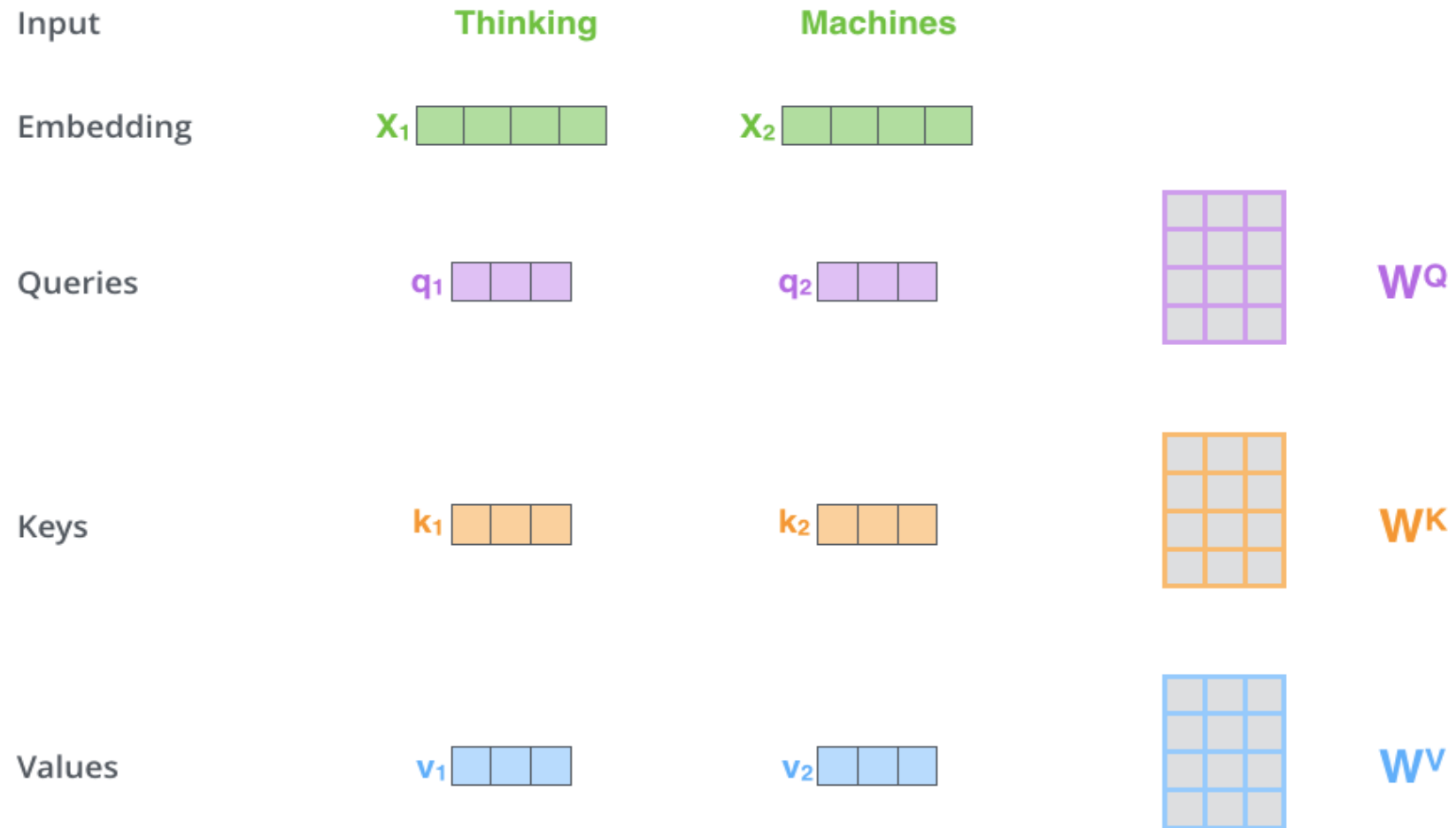


$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

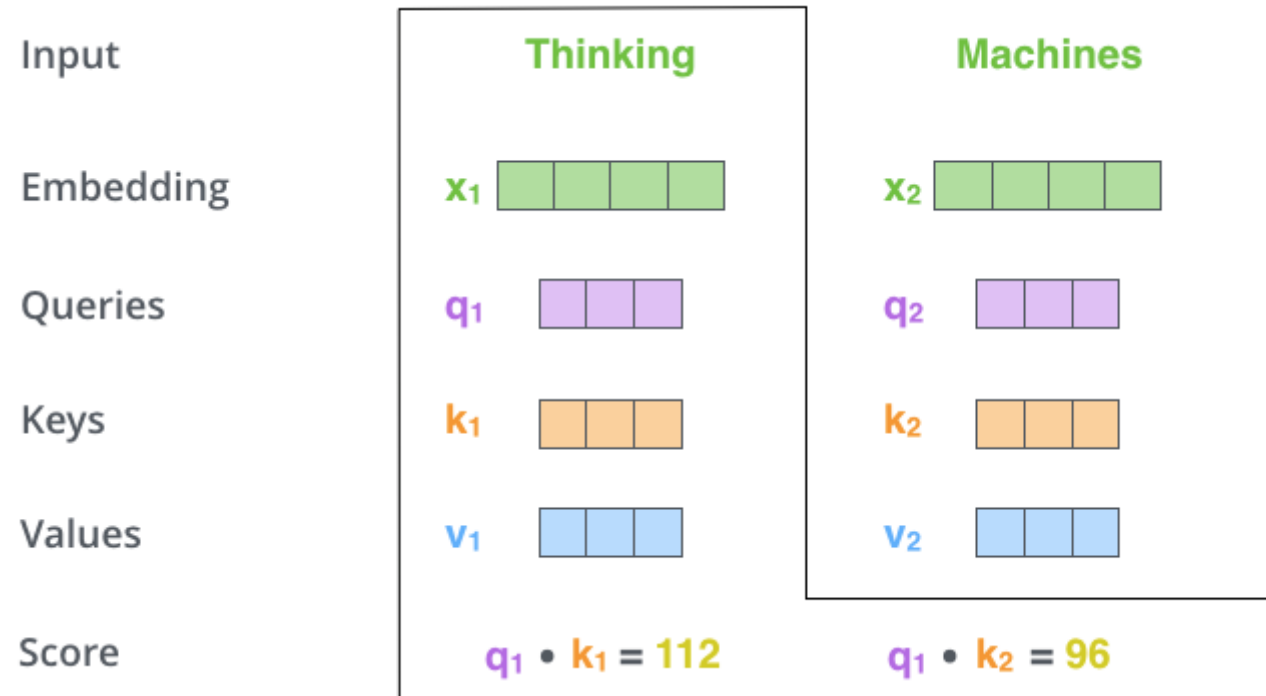
Mécanisme de self-attention



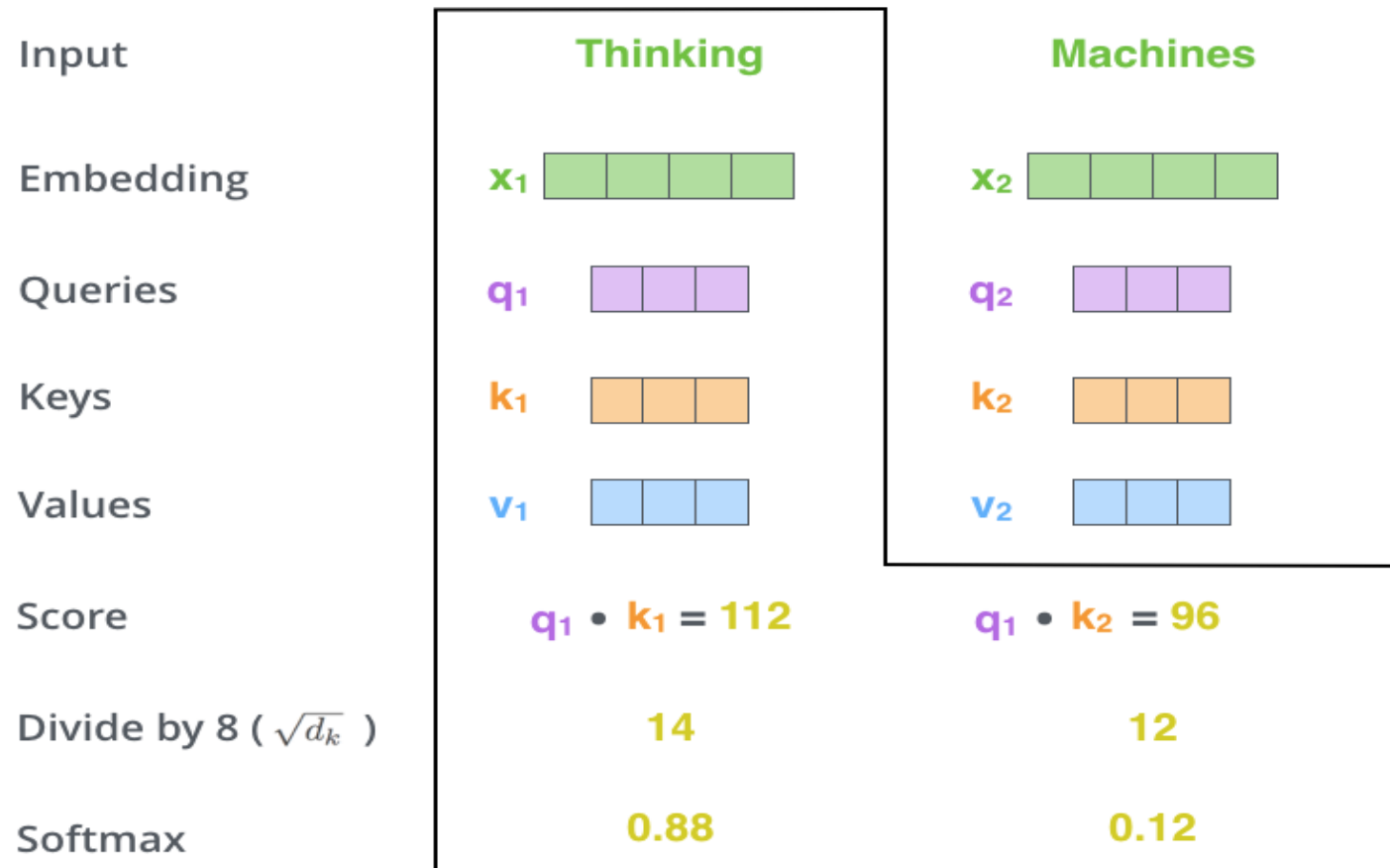
Mécanisme de self-attention



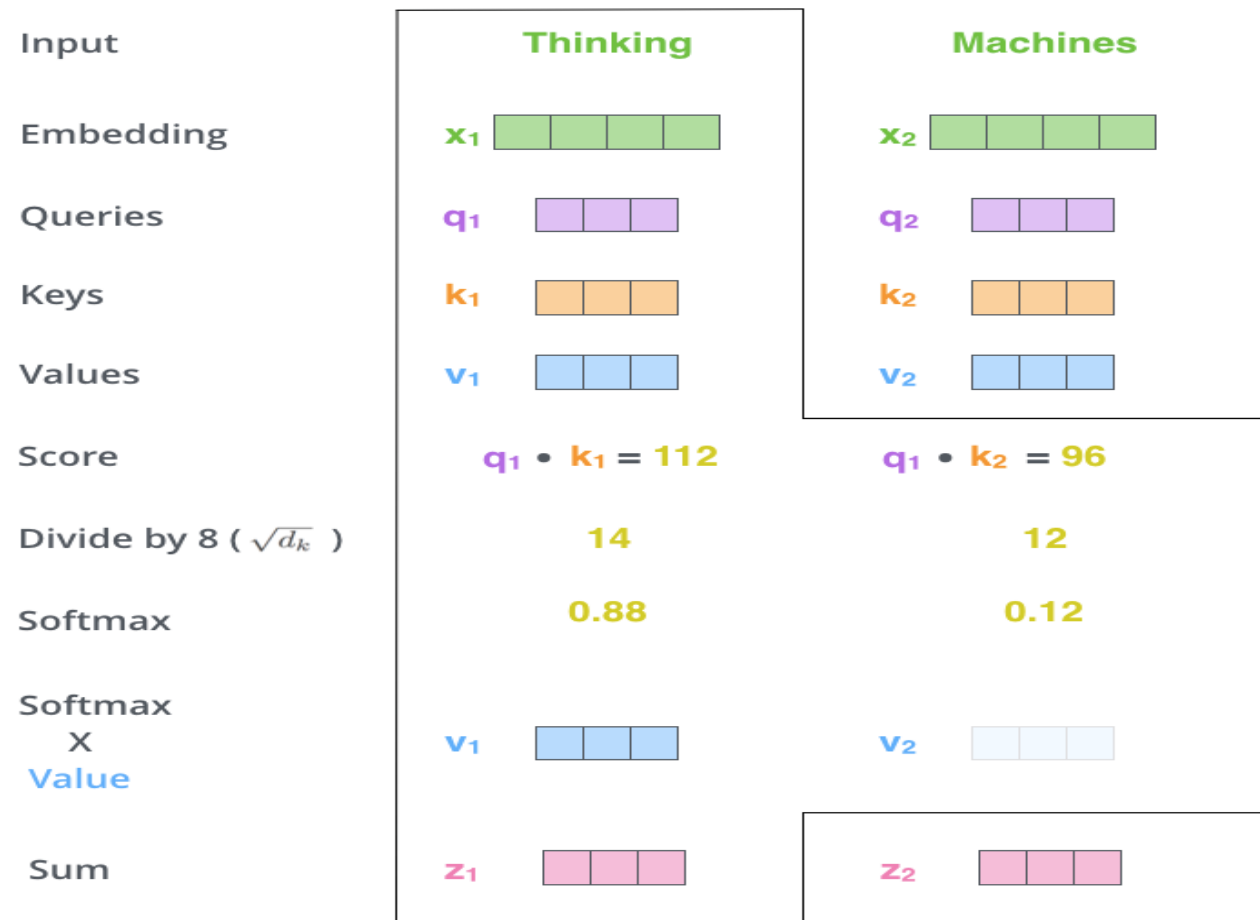
Mécanisme de self-attention




Mécanisme de self-attention



Mécanisme de self-attention



Mécanisme de self-attention : formulation matricielle

$$\mathbf{X} \times \mathbf{W}^Q = \mathbf{Q}$$


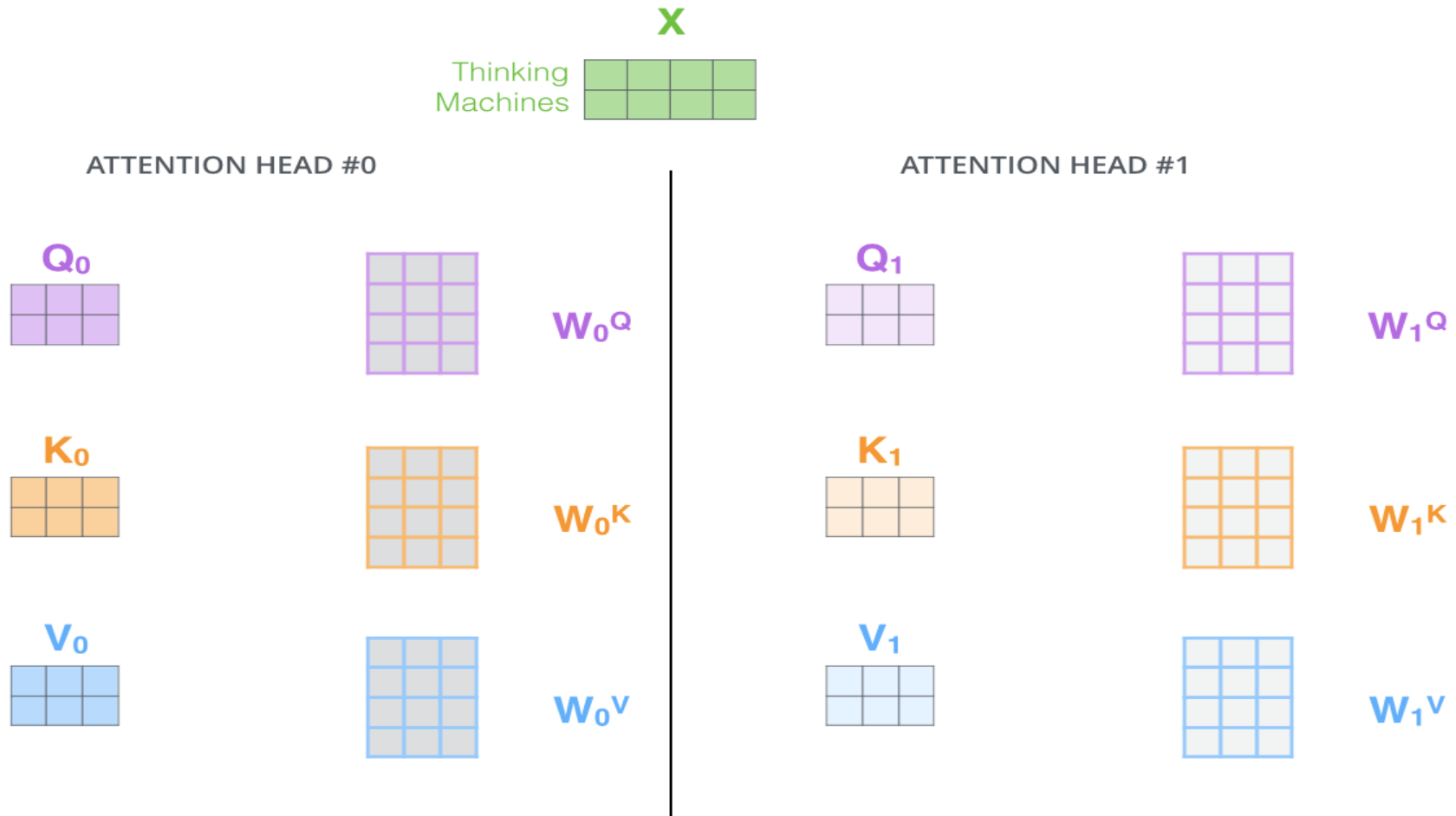
$$\mathbf{X} \times \mathbf{W}^K = \mathbf{K}$$


$$\mathbf{X} \times \mathbf{W}^V = \mathbf{V}$$

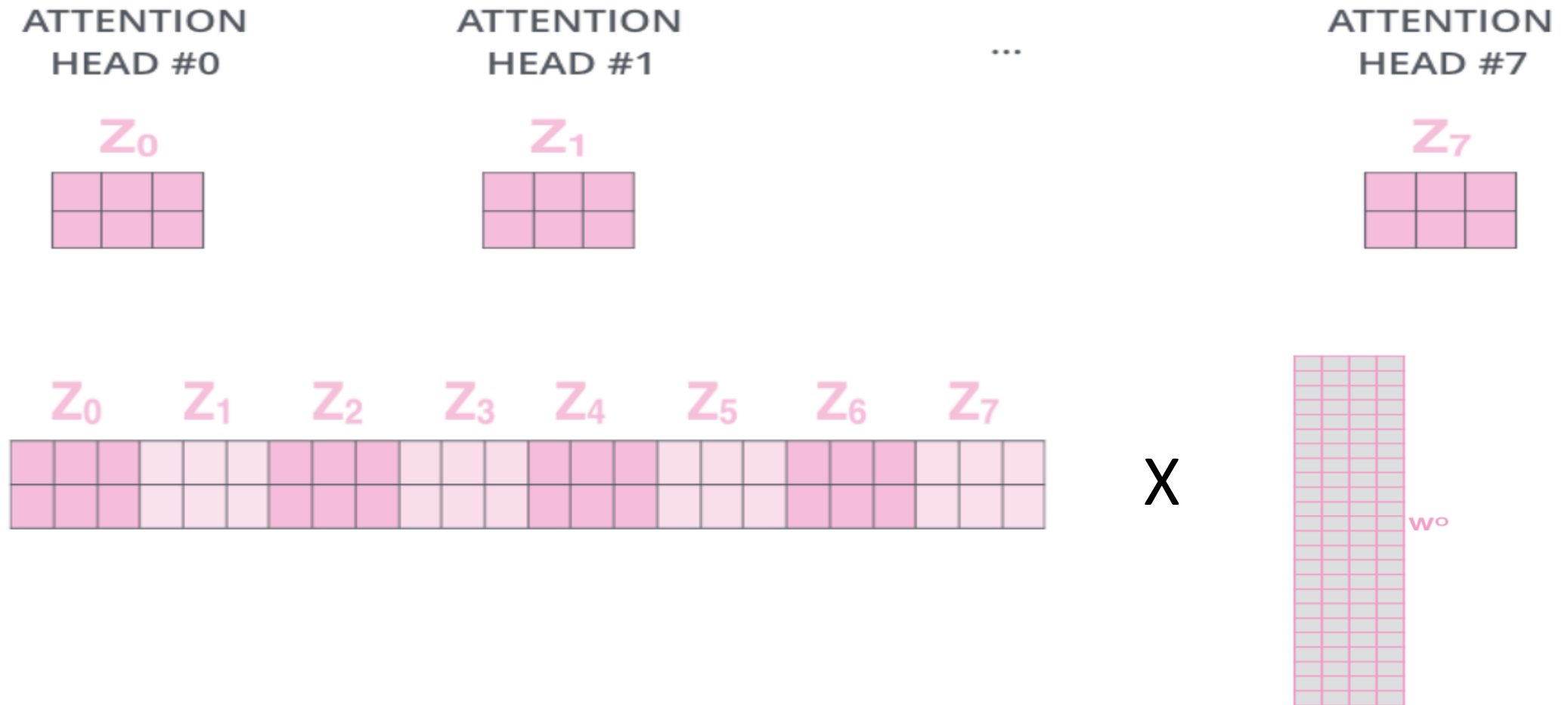

Mécanisme de self-attention : formulation matricielle

$$\text{softmax}\left(\frac{\begin{matrix} \text{Q} \\ \begin{array}{|c|c|c|} \hline & & \\ \hline \end{array} \end{matrix} \times \begin{matrix} \text{K}^T \\ \begin{array}{|c|c|} \hline & \\ \hline \end{array} \end{matrix}}{\sqrt{d_k}}\right) \begin{matrix} \text{V} \\ \begin{array}{|c|c|c|} \hline & & \\ \hline \end{array} \end{matrix}$$
$$= \begin{matrix} \text{Z} \\ \begin{array}{|c|c|c|} \hline & & \\ \hline \end{array} \end{matrix}$$

Multihead Attention



Multihead Attention



Performances du Transformer originel

Entraîné sur WMT 2014
English-

German dataset, comprenant
près de 4.5 millions de
phrases sentence, le WMT
2014 English-French dataset,
comprenant près de 36
millions de phrases.

8 NVIDIA P10 GPUs

Base: 12 heures

Big: 3.5 jours

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	41.29	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$3.3 \cdot 10^{18}$	
Transformer (big)	28.4	41.8	$2.3 \cdot 10^{19}$	

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova

Google AI Language

{jacobdevlin, mingweichang, kentonl, kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

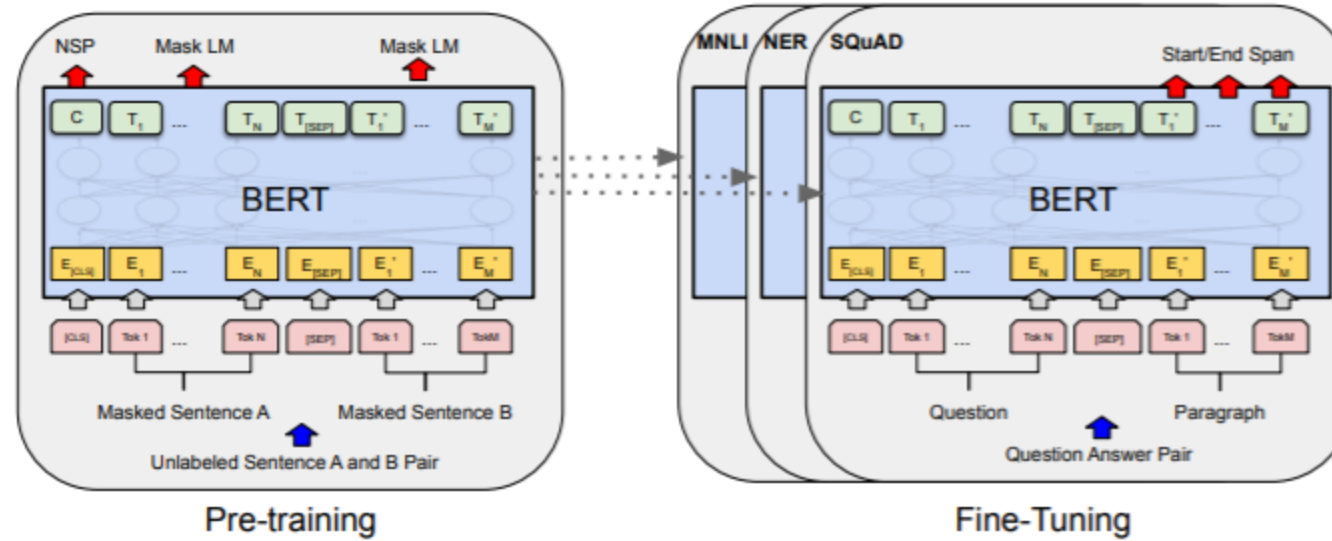
There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

We argue that current techniques restrict the power of the pre-trained representations, especially for the fine-tuning approaches. The major limitation is that standard language models are unidirectional, and this limits the choice of architectures that can be used during pre-training. For example, in OpenAI GPT, the authors use a left-to-right architecture, where every token can only attend to previous tokens in the self-attention layers of the Transformer (Vaswani et al., 2017). Such restrictions are sub-optimal for sentence-level tasks,

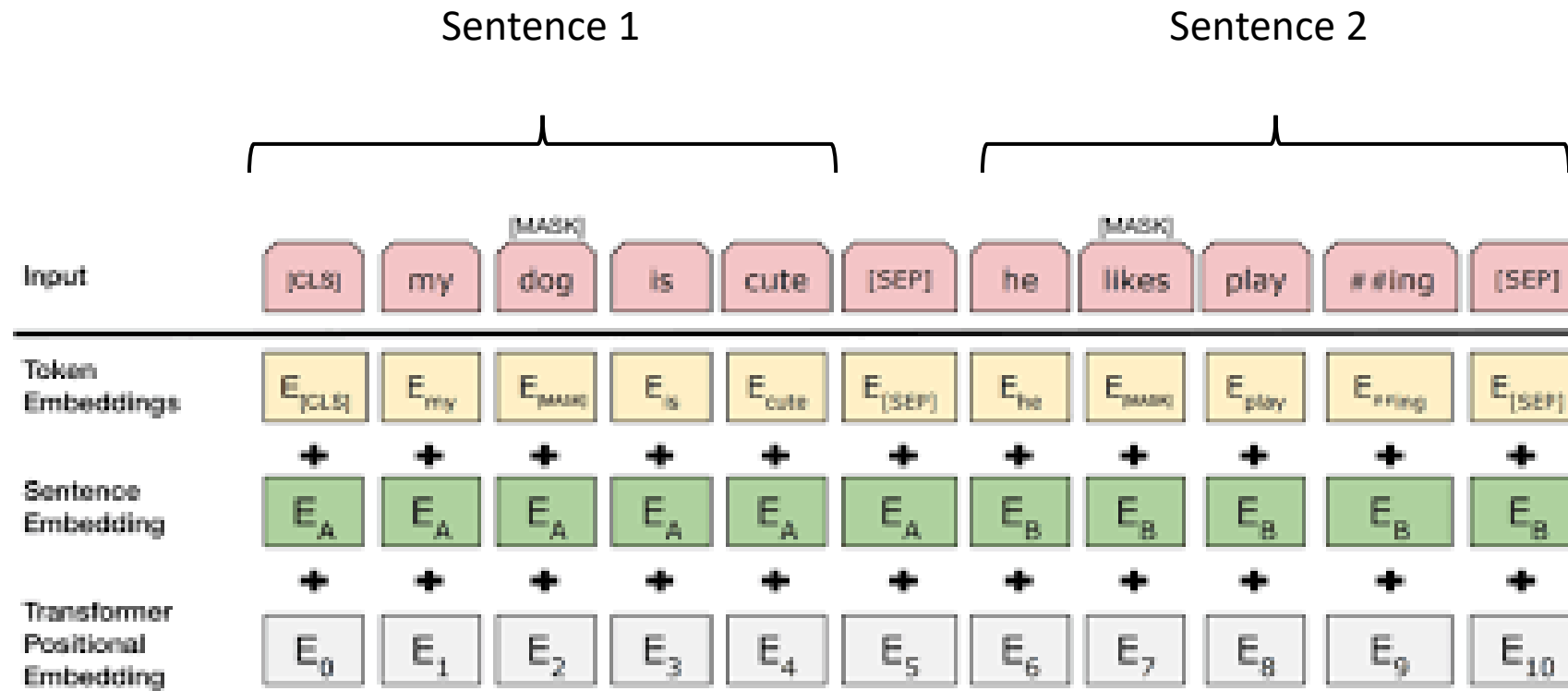
v:1810.04805v2 [cs.CL] 24 May 2019

1 Introduction

BERT : pre-training et fine-tuning



BERT : pre-training



BERT : architecture et performance

BERT a été pré-entraîné sur le BookCorpus (800 millions de mots) et English Wikipedia (2,500 millions de mots).

Deux modèles :

- BERT Base: 12 couches avec 110 millions de paramètres.
- BERT Large: 24 couches avec 340 millions de paramètres.

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
BERT _{BASE}	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT _{LARGE}	86.7/85.9	72.1	92.7	94.9	60.5	86.5	89.3	70.1	82.1

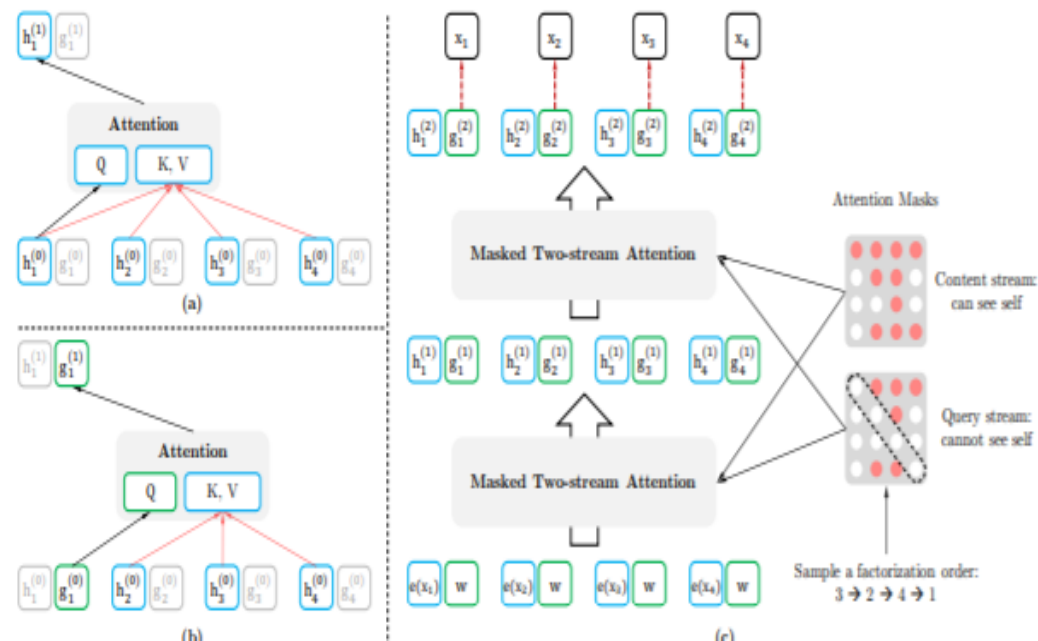
XLNet

To better understand the difference, let's consider a concrete example [New, York, is, a, city]. Suppose both BERT and XLNet select the two tokens [New, York] as the prediction targets and maximize $\log p(\text{New York} \mid \text{is a city})$. Also suppose that XLNet samples the factorization order [is, a, city, New, York]. In this case, BERT and XLNet respectively reduce to the following objectives:

$$\mathcal{J}_{\text{BERT}} = \log p(\text{New} \mid \text{is a city}) + \log p(\text{York} \mid \text{is a city}),$$

$$\mathcal{J}_{\text{XLNet}} = \log p(\text{New} \mid \text{is a city}) + \log p(\text{York} \mid \text{New, is a city}).$$

Notice that XLNet is able to capture the dependency between the pair (New, York), which is omitted by BERT. Although in this example, BERT learns some dependency pairs such as (New, city) and (York, city), it is obvious that XLNet always learns **more** dependency pairs given the same target and contains “denser” effective training signals.



XLNet : architecture et performance

XLNet a la même architecture que BERT Large.

Model	SQuAD1.1	SQuAD2.0	RACE	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B
BERT-Large (Best of 3)	86.7/92.8	82.8/85.5	75.1	87.3	93.0	91.4	74.0	94.0	88.7	63.7	90.2
XLNet-Large- wikibooks	88.2/94.0	85.1/87.8	77.4	88.4	93.9	91.8	81.2	94.4	90.0	65.2	91.1

GPT-3 : few shot learner

The three settings we explore for in-context learning

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 cheese => ..... ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← example
3 cheese => ..... ← prompt
```

Few-shot

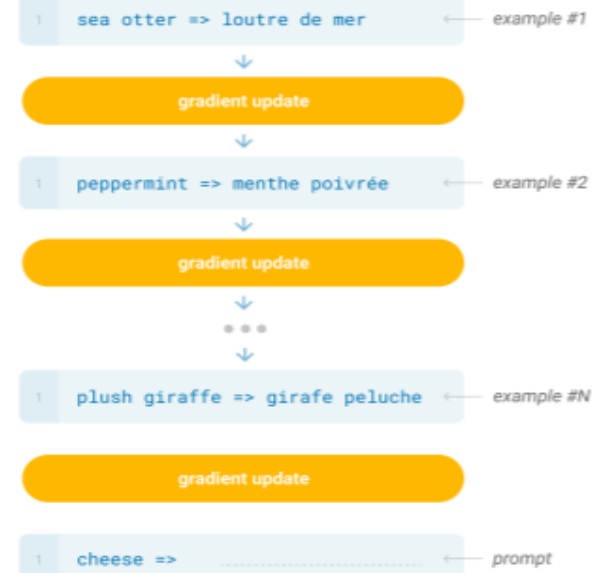
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← examples
3 peppermint => menthe poivrée ←
4 plush girafe => girafe peluche ←
5 cheese => ..... ← prompt
```

Traditional fine-tuning (not used for GPT-3)

Fine-tuning

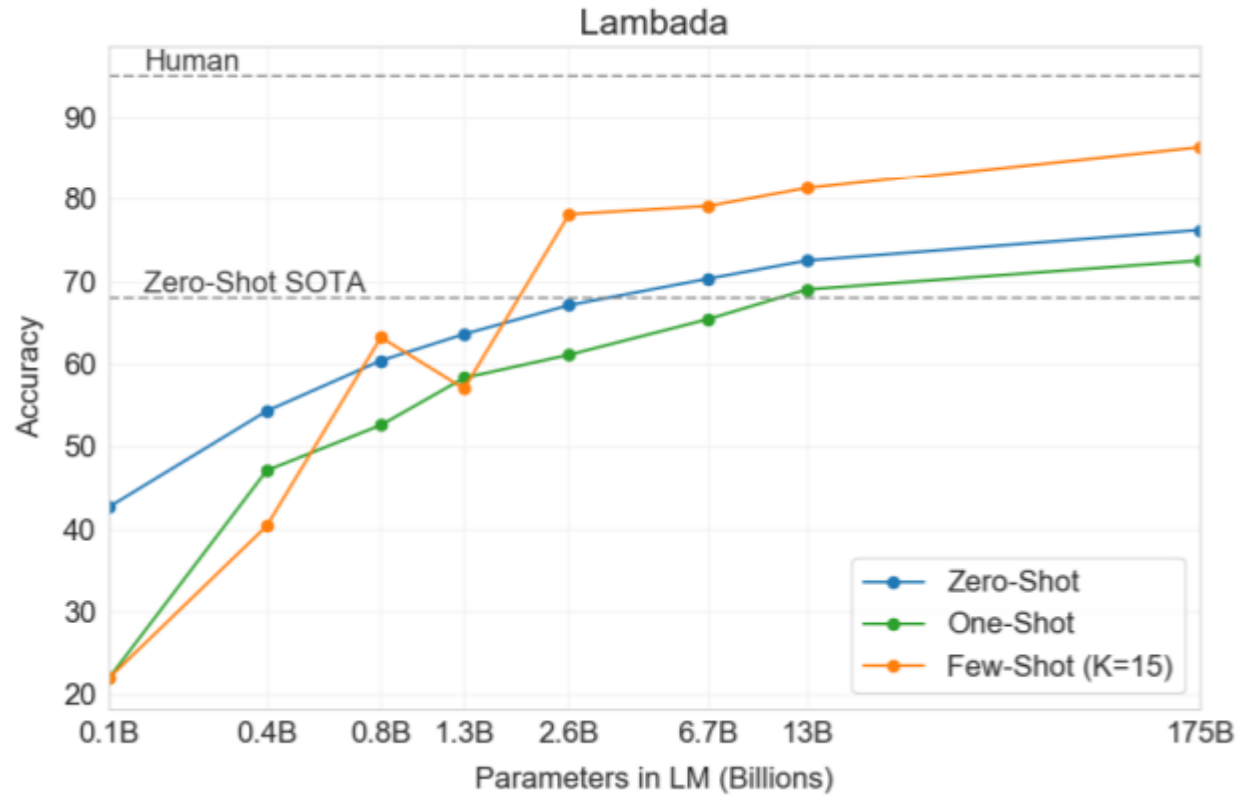
The model is trained via repeated gradient updates using a large corpus of example tasks.



GPT-3 : architecture et performance

Le modèle a été entraîné sur Common Crawl dataset et WebText2, Books1, Books2 et Wikipedia, pour un total de près de 500 milliards de tokens.

Le modèle GPT-3 le plus large possède **96 couches** et **175 billion paramètres**.



GPT-3 : exemple de génération de texte

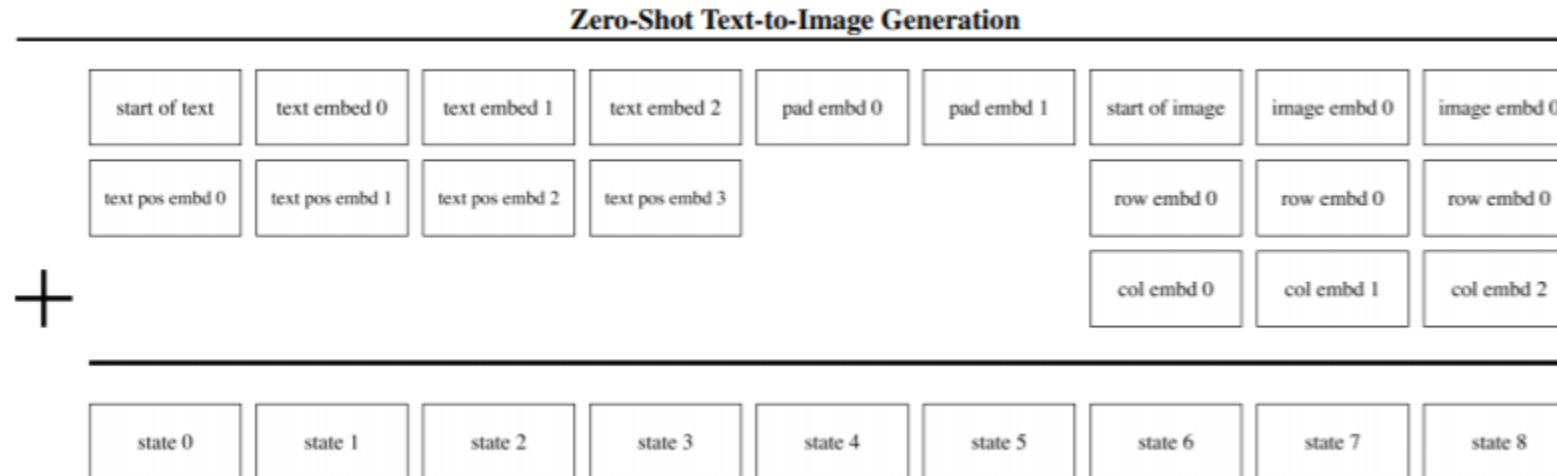
----- Generated Poem 1 -----

I must have shadows on the way
If I am to walk I must have
Each step taken slowly and alone
To have it ready made

And I must think in lines of grey
To have dim thoughts to be my guide
Must look on blue and green
And never let my eye forget
That color is my friend
And purple must surround me too

The yellow of the sun is no more
Intrusive than the bluish snow
That falls on all of us. I must have
Grey thoughts and blue thoughts walk with me
If I am to go away at all.

DALL-e : apprentissage multimodal



DALL-e : génération d'images

TEXT PROMPT

an armchair in the shape of an avocado [...]

AI-GENERATED IMAGES



TEXT PROMPT

a store front that has the word 'openai' written on it [...]

AI-GENERATED IMAGES



Merci de votre attention

Q&A

Annexe

TF-IDF avec Python

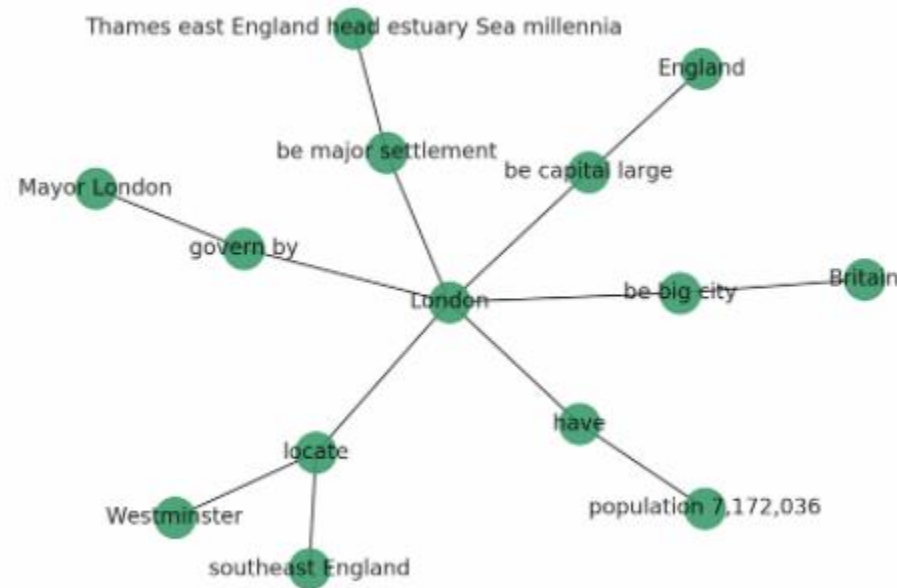
Ce document présente une implémentation de TF-IDF simple et rapide avec scikit-learn (Python).

<https://programmerbackpack.com/tf-idf-explained-and-python-implementation/>*

Extraction d'informations et graphes de connaissances

Un document qui détaille à la fois l'extraction d'information et leur représentation graphique (très utile) avec Python.

<https://programmerbackpack.com/python-nlp-tutorial-information-extraction-and-knowledge-graphs/>



Latent Dirichlet Allocation (LDA)

Un document expliquant l'algorithme LDA, la principale technique utilisée pour le topic modeling, d'une manière schématique.

<https://www.mygreatlearning.com/blog/understanding-latent-dirichlet-allocation/>

Pour les détails mathématiques, on peut se référer au papier scientifique originel.

https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf?TB_iframe=true&width=370.8&height=658.8

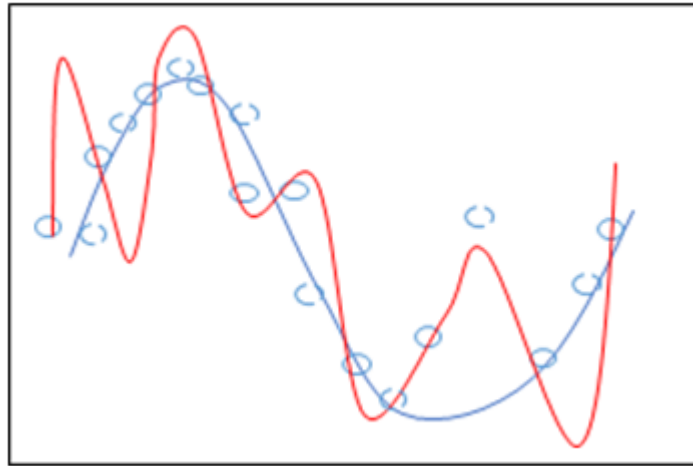
A noter que le topic modeling peut également être effectuée avec des techniques de Deep Learning, notamment les Transformers. Ci-dessous un document Colab exhaustif sur le Topic Modeling avec BERTopic.

<https://colab.research.google.com/drive/1FieRA9fLdkQEGDIMYlOI3MCjSUKVF8C-?usp=sharing>

Surapprentissage (overfitting) et régularisation

Une très bonne illustration de la notion de surapprentissage en machine learning avec un rappel des principales techniques de régularisation pour y remédier. Il s'agit notamment de L1 et L2. Pour tout ce qui est relatif au deep learning, il faut plutôt se référer aux techniques de dropout et de early stopping.

<https://blogs.sas.com/content/subconsciousmusings/2017/07/06/how-to-use-regularization-to-prevent-model-overfitting/>



Problème des données déséquilibrées

Le problème de données déséquilibrées apparaît dans les modèles de classification lorsqu'il y a une classe (ou plusieurs classes) largement majoritaire par rapport aux autres. Les exemples les plus courants sont la fraude aux cartes bancaires, le ciblage marketing (peu de gens ouvrent les mails publicitaires), les maladies rares, etc.

La plupart des modèles de classification sont très sensibles aux datasets déséquilibrés, notamment la régression logistique et les SVMs. Il existe certaines techniques permettant de remédier à ce problème, détaillées dans le document suivant (leur efficacité est en réalité très relative).

<https://www.kdnuggets.com/2019/05/fix-unbalanced-dataset.html>

Concept drift

Le concept drift est un problème très courant en machine learning. Il apparaît dans le contexte de la mise en production d'un modèle, lorsque la distribution des données change par rapport à celle sur laquelle le modèle en question a été entraîné. Ce document explique d'une manière assez détaillée ce problème et rappelle les principales techniques pour y remédier.

<https://machinelearningmastery.com/gentle-introduction-concept-drift-machine-learning/>

Fonctions coût (loss functions)

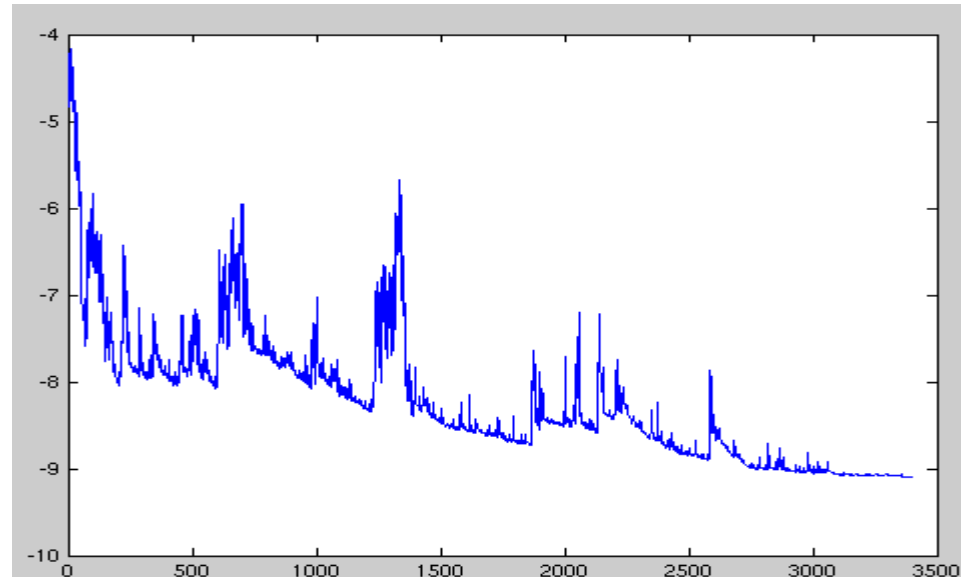
L'entraînement des modèles de machine learning impliquent généralement la minimisation d'une fonction coût (erreur). Il existe deux catégories de fonctions coût pour l'apprentissage supervisé : les fonctions coût pour la régression et les fonctions coût pour la classification. Le document suivant rappelle les fonctions coût les plus courantes.

<https://www.section.io/engineering-education/understanding-loss-functions-in-machine-learning/>

Descente stochastique du gradient

Les paramètres des modèles de machine learning, notamment les réseaux de neurones, sont optimisés à l'aide de techniques de type descente stochastique du gradient (SGD). Il existe plusieurs variantes, rappelées d'une manière exhaustive dans le document suivant. La plus utilisée pour les réseaux de neurones profonds reste néanmoins Adam.

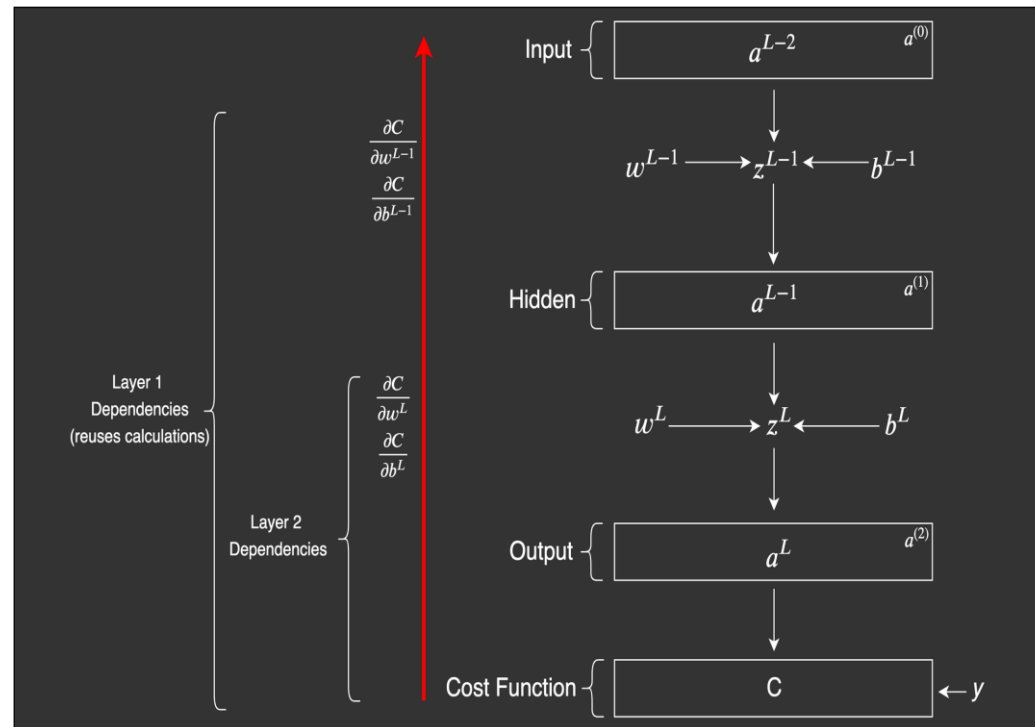
<https://runder.io/optimizing-gradient-descent/>



Backpropagation

Ce document explique en détails la procédure d'entraînement des réseaux de neurones, notamment la rétropropagation du gradient (backpropagation).

<https://mlfromscratch.com/neural-networks-explained/#/>

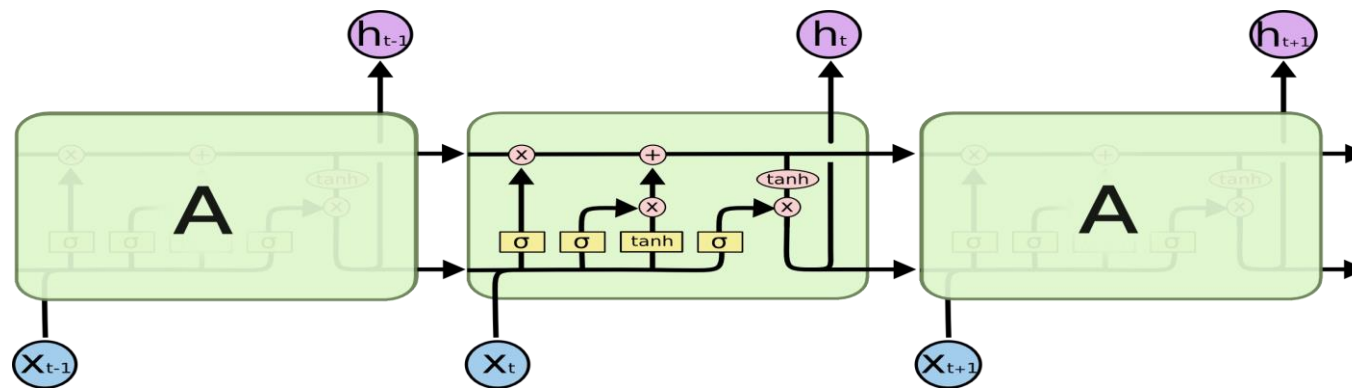


LSTM

Les LSTMs sont parmi les modèles de deep learning les plus utilisés (encore aujourd'hui) pour le traitement des données séquentielles (séries temporelles, texte, etc.).

Le document suivant explique leur fonctionnement, étape par étape, d'une manière assez détaillée.

<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>



Mécanisme d'Attention

Ce document, en français, détaille le mécanisme d'Attention sur un exemple de NMT (Neural Machine Translation). Il est très clair et plutôt facile à lire. Le mécanisme d'Attention est la notion clé dans le deep learning moderne, notamment en NLP.

https://medium.com/@pierre_guillou/nlp-fastai-attention-mechanism-3c35ac3de5b9

Transformers

Les Transformers sont les modèles de deep learning à l'état de l'art, notamment en NLP. Leur fonctionnement repose sur le mécanisme d'Attention. Ce document les explique d'une manière intuitive et schématique.

<https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04>

Pour davantage de détails mathématiques, se référer au papier originel introduisant les Transformers « Attention is all you need ».

<https://arxiv.org/pdf/1706.03762.pdf>

BERT

BERT est l'un des modèles de langage les plus puissants et les plus versatiles. Il constitue en soi une révolution dans le domaine du NLP. Ce document explique les principales caractéristiques du modèle BERT et sa procédure de fine-tuning.

<https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270>

BERT a donné lieu à une multitude de variantes, dont une bonne partie est listée sur le site de Hugging Face (entre autres).

https://huggingface.co/transformers/pretrained_models.html

GPT-3

GPT-3 est aujourd'hui le modèle le plus puissant en NLP. Il peut s'adapter à certaines tâches en ayant vu un nombre très limité d'exemple (une quinzaine typiquement).

Le document suivant détaille le fonctionnement de GPT-3 et rappelle son histoire à travers GPT et GPT-2.

<https://medium.com/walmartglobaltech/the-journey-of-open-ai-gpt-models-32d95b7b7fb2>

Cet article, d'OpenAI, liste les applications déjà concrètes de GPT-3, déployé en version bêta.

<https://openai.com/blog/gpt-3-apps/>

Apprentissage multimodal

L'apprentissage multimodal est le sujet brûlant (hot topic) du deep learning moderne. Il consiste à entraîner un modèle sur différentes typologies de données (images, texte, etc.). Pour certains, il s'agit même d'un chemin prometteur vers l'intelligence artificielle générale (AGI).

Ce document rapporte les dernières expérimentations et trouvailles d'OpenAI sur l'apprentissage multimodal, en utilisant un modèle de type GPT-3 (CLIP).

<https://openai.com/blog/clip/>

Références générales sur le NLP

- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc."
- Hapke, H., Howard, C., & Lane, H. (2019). *Natural Language Processing in Action: Understanding, analyzing, and generating text with Python*. Simon and Schuster.
- Rao, D., & McMahan, B. (2019). *Natural language processing with PyTorch: build intelligent language applications using deep learning*. " O'Reilly Media, Inc."