Introduction au **traitement automatique du langage naturel** (TALN ou NLP)

GAUTIER DURANTIN

ISAE-SUPAÉRO 2020-2021

GAUTIER.DURANTIN@E-I.COM

Contexte : comprendre le langage

Les 4 constituants d’une langue

1. Le langage est **CULTUREL**

= production influencée par des facteurs de contexte, mais également sociologiques, économiques, physiologiques, etc…
= **difficulté d’implémenter des solutions programmées pour son traitement**

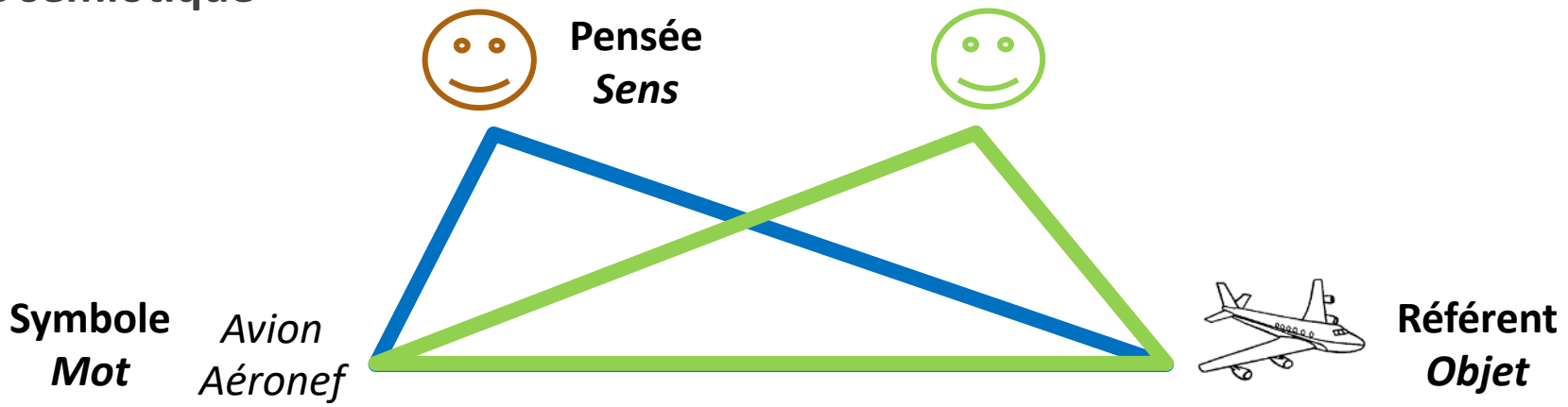
Les 4 constituants d'une langue

1. Le langage est **CULTUREL**

= production influencée par des facteurs de contexte, mais également sociologiques, économiques, physiologiques, etc…
= **difficulté d'implémenter des solutions programmées pour son traitement**

2. L'**attention jointe** (joint attention)

= **triangle sémiotique**



Les 4 constituants d'une langue

3. Le **déplacement** (displacement)

= capacité à faire référence à des objets ***distants*** ou ***abstraits***

= l'objet auquel on fait référence (*référent*) dans un énoncé (*utterance*) sont souvent inaccessibles dans la donnée autrement que par celui-ci.

Les 4 constituants d'une langue

3. Le **déplacement** (displacement)

= capacité à faire référence à des objets ***distants*** ou ***abstraits***

= l'objet auquel on fait référence (*référent*) dans un énoncé (*utterance*) sont souvent inaccessibles dans la donnée autrement que par celui-ci.

2. L'**infinité finie**, ou **productivité**

= avec un nombre fini de symboles, il est possible d'exprimer un nombre infini de concepts.
= on peut produire à l'infini des façons différentes d'exprimer une seule idée

« il fera beau demain », « je dis qu'il fera beau demain », « elle dit que je dis qu'il fera beau demain », ….

La difficulté d'extraire le sens

SEMANTIQUE : le sens des mots

« ***Continuez comme ça, c'est super !*** »

VS

PRAGMATIQUE : l'intention derrière les mots

« Nous avons été au NOM_DU_RESTAURANT à l'occasion de notre anniversaire de mariage. Quelle déception ! Serveur aimable comme une porte de prison, plats servis froids, et attente interminable. ***Continuez comme ça, c'est super !*** »

Maximes de Grice et pragmatique

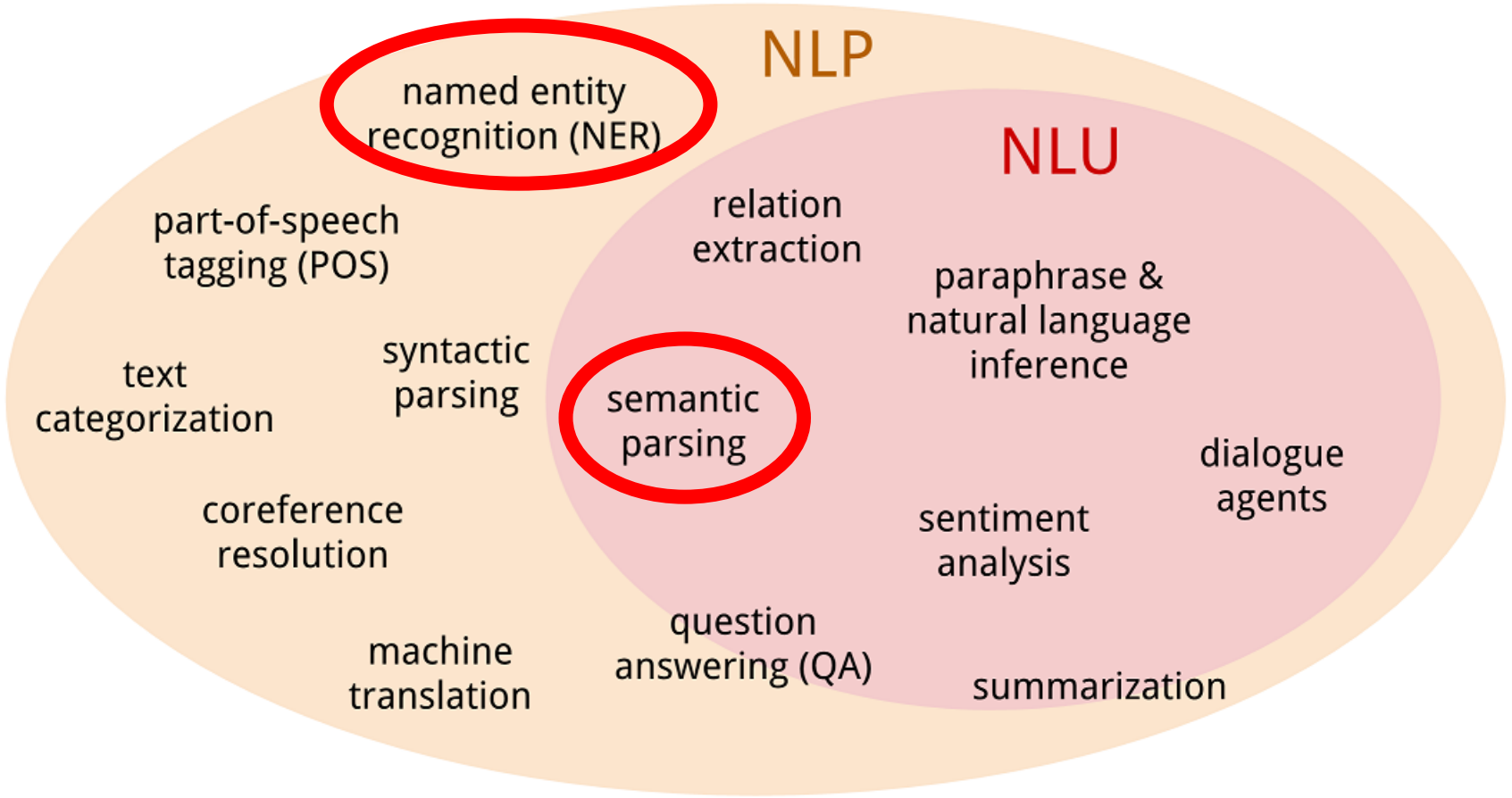
La condition pour que **la sémantique et la pragmatique** d'un énoncé soient identiques repose sur 4 maximes :

- **Quantité :** l'énoncé doit contenir autant d'information que nécessaire. Pas plus, pas moins.
- **Qualité :** je ne dois pas dire ce que je pense être faux, ou ce que je n'ai pas de raison suffisante de considérer comme vrai
- **Relation :** je dois être pertinent (e.g. ne pas changer de sujet, répondre à coté, etc…)
- **Manière :** mes énoncés doivent être brefs, sans ambiguïté et ordonnés

A chaque infraction aux maximes de Grice, il se peut que la sémantique et la pragmatique d'un énoncé diffèrent. Le message véhicule alors une **intention** particulière, et on parle d'énoncé **performatif**.

Les grandes familles de problèmes en NLP

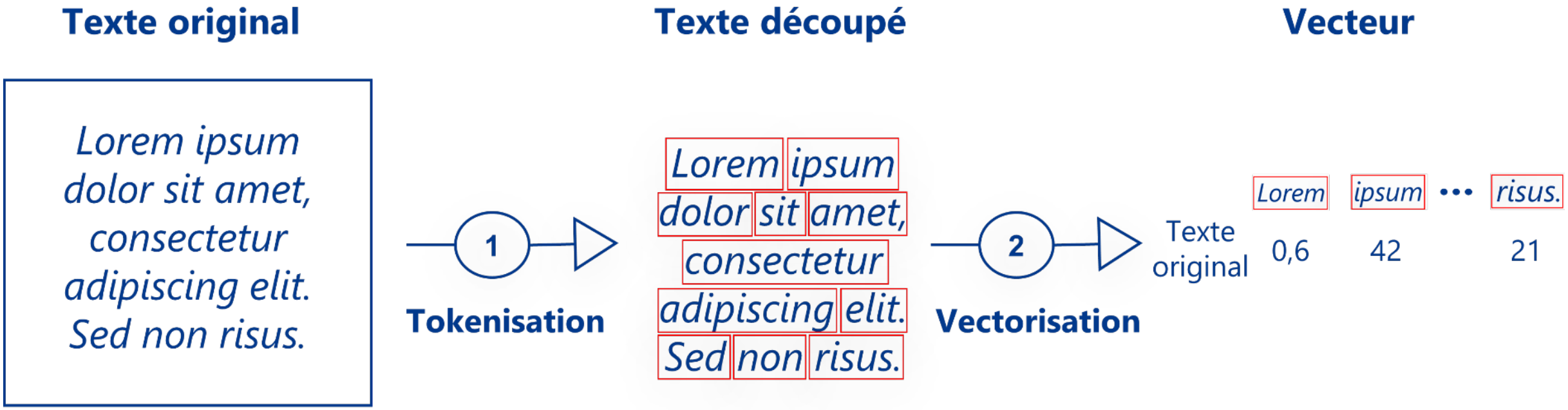
Les grandes familles de problèmes en NLP



La majorité des problèmes se ramène en général à un problème de classification ou d'extraction d'entités

Prétraitements

Tokenisation et Vectorisation



Pour faciliter ces deux étapes, on effectue souvent un **Prétraitement** du texte original en amont.

Prétraitements communs à sélectionner

Traitements typographiques

- **lowercasing** : uniformisation de la casse d'un document
- **normalisation :** remplacement/retrait de ponctuations et caractères spéciaux (accents, symboles…)

Traitements morphologiques (affectent la forme des mots et donc la lisibilité de la phrase)

- **Lemmatisation :** un mot est remplacé par son lemme (e.g ***aurions*** devient ***avoir***)
  - Problème : la lemmatisation est une tâche compliquée et coûteuse (par exemple : « *nous avions des avions* »). Elle doit s'appuyer sur la connaissance de **la nature et de la fonction** du token pour être performante, ce qui requiert une analyse grammaticale (Part Of Speech tagging)
  - La complexité de la lemmatisation est d'autant plus grande que la richesse de flexion de la langue est grande (conjugaisons, accords, etc…)
- **Racinisation (stemming) :** le but est le même que la lemmatisation, mais on utilise des règles pour tronquer un mot à sa racine.
  - Exemple : « *je souhaite fermer mon compte* » devient « *je souhait ferm mon compt* »
- Retrait de « **stopwords** » : mots indispensables à la structure de la phrase, mais ne portant pas de sens important (e.g. pour, de, le, la, je, il, qui …..) => un prétraitement commun peut consister à ne conserver que les noms, adjectifs, adverbes et verbes.

Tokenisation

Des règles simples à mettre en place automatiquement :

- Séparation des mots par les espaces.
- Suppression de la ponctuation.

« Mercredi, il y a des frites. »

« Mercredi » « il » « y » « a » « des » « frites »
6 tokens

Mais :

« Aujourd'hui, il y a peut-être des frites. »

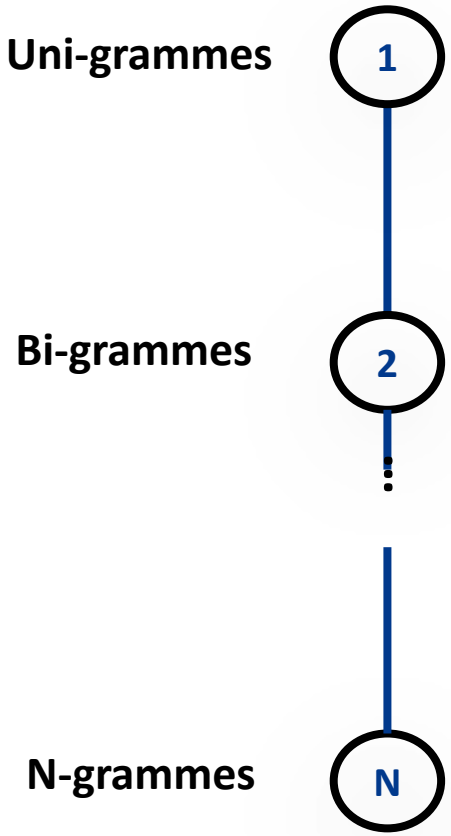
« Aujourd » « hui » « il » « y » « a »
« peut » « être » « des » « frites »

« Je le sais dores et déjà. »

« Je » « le » « sais » « dores »
« et » « déjà »

On aimerait un seul token pour : « Aujourd hui » « peut être » « dores et déjà »

Tokenisation – modèles n-gram



« Mercredi, il y a peut-être des frites. »

« Mercredi » « il » « y » « a » « peut » « être » « des » « frites »
8 tokens

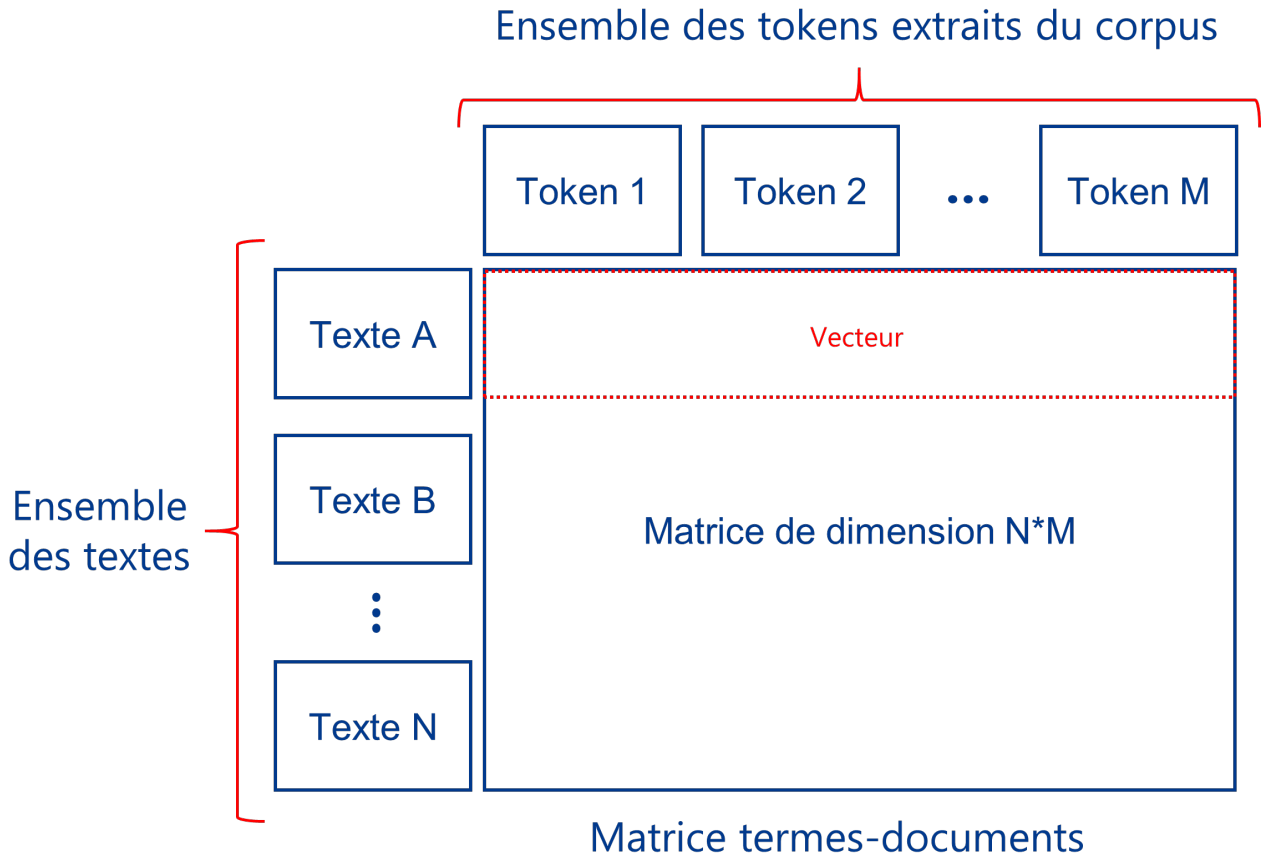
« Mercredi, il y a peut-être des frites. »

« Mercredi » « Mercredi il » « il » « il y » « y » « y a » « a » « a peut » « peut » « peut être » « être » « être des » « des » « des frites » « frites »
15 tokens

La tokenisation avec utilisation d'un modèle n-gram permet d'augmenter la capacité à capturer l'importance d'expressions de plusieurs mots, mais multiplie le nombre de tokens, et donc la complexité du modèle

Vectorization

BAG OF WORDS (BOW)



Un token peut être une lettre, une chaine de caractères ou un mot

$$tokens = \{t_i, i \in [\![1, M]\!]\}$$

$$documents = \{d_j, j \in [\![1, N]\!]\}$$

$$TF(j,i) = fréquence\ d'apparition\ du\ token\ t_i\ dans\ le\ document\ d_j$$

La représentation des documents en bag of words est simple, mais :

- Biaisée pour les longs documents (TF -> 0)
- Donne trop de poids aux mots très courants et portant peu de sens (e.g. articles définis, pronoms personnels)

Term Frequency - Inverse Document Frequency

$tokens = \{t_i, i \in [\![1, M]\!]\}$

$documents = \{d_j, j \in [\![1, N]\!]\}$

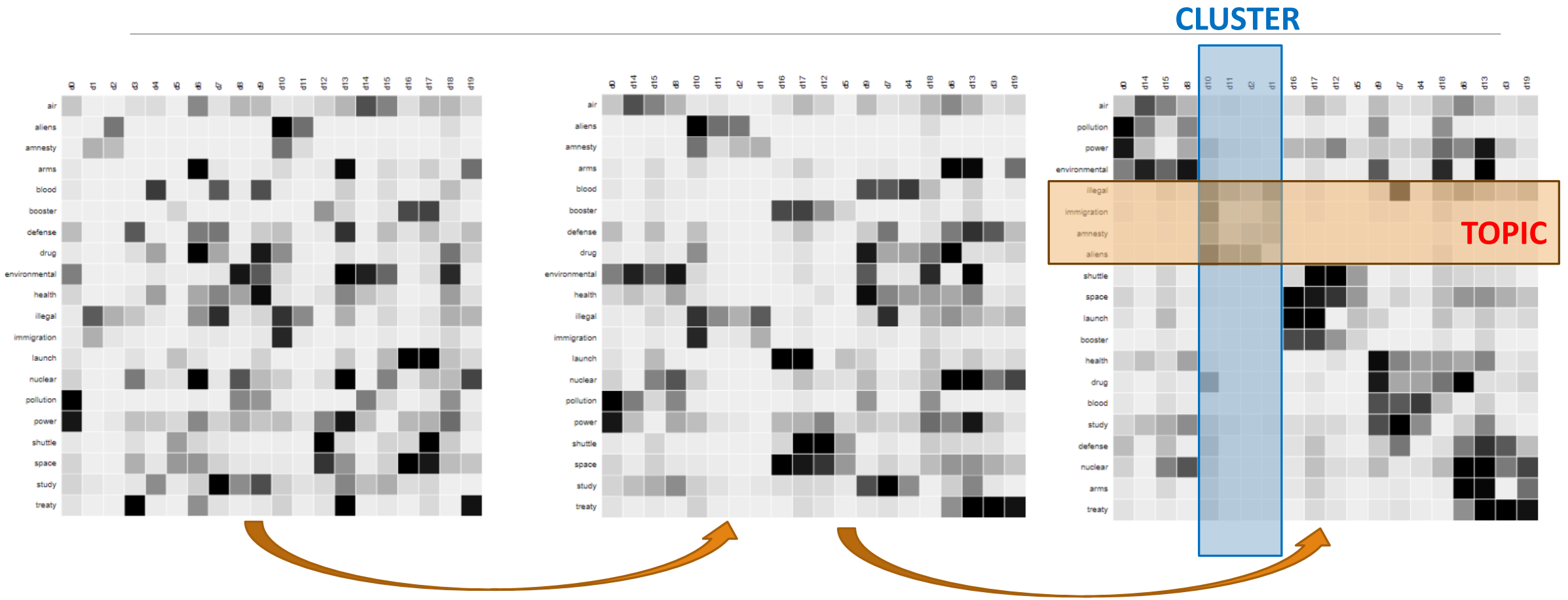
$TF(j, i) = fréquence\ d'apparition\ du\ token\ t_i\ dans\ le\ document\ d_j$

$DF(i) = Nombre\ de\ documents\ du\ corpus\ contenant\ le\ token\ t_i$

$$\boldsymbol{TFIDF_{t,d}} = TF_{t,d} \times \log\left(\frac{Nombre\ total\ de\ document}{DF_t}\right) = TF_{t,d} \times IDF_t$$

Tend vers 0 pour les termes communs aux documents du corpus

Latent Semantic Analysis (LSA) – topic modeling



Groupement par similarité entre documents (colonnes)

Groupement par similarité entre termes (lignes)

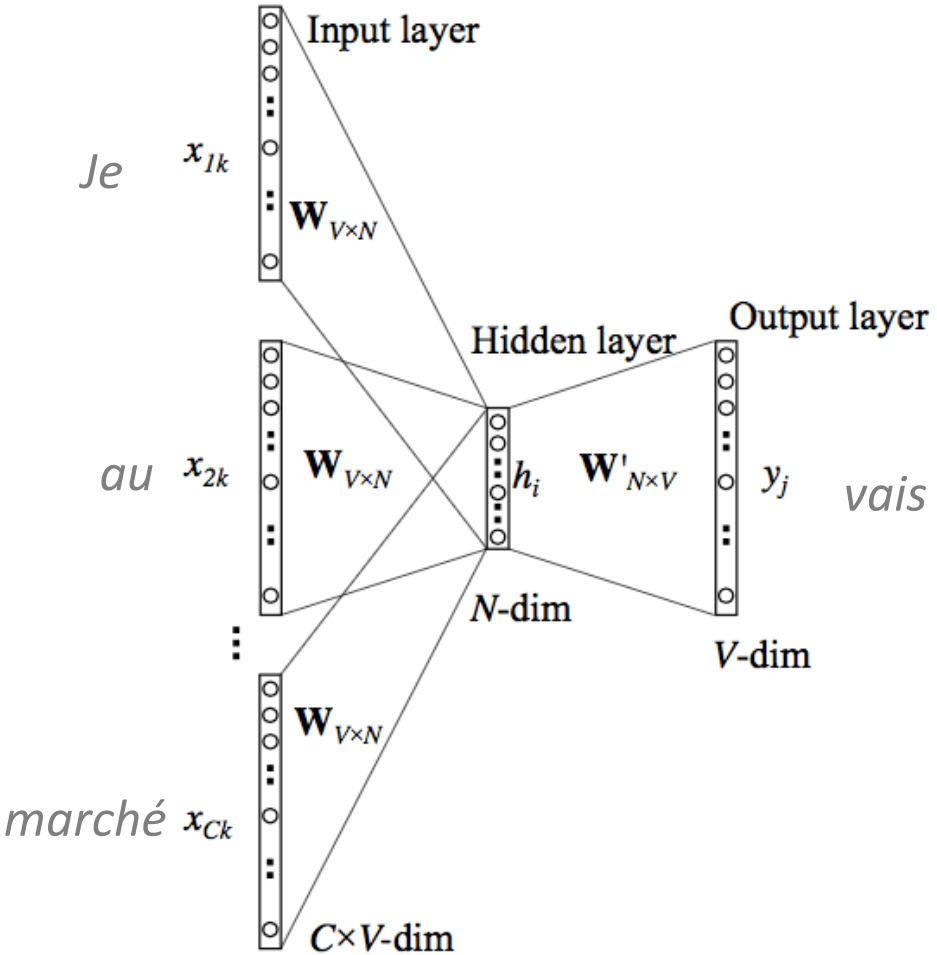
Les limites des approches BOW et TFIDF

- Sur une tache simple et dans un contexte linguistique contrôlé (=taille de vocabulaire réduite), les méthodes TFIDF et BOW sont **généralement les plus performantes**, pour des tâches de classification

- Lorsque la taille du vocabulaire augmente, la **dimensionnalité de l'espace de vectorisation** aussi => Perte de performance (même si ce sont des vecteurs sparse)

- Une alternative consiste à associer à chaque mot du vocabulaire un unique vecteur de réels de dimension fixée (représentation du document par une combinaison des vecteurs des mots le composant). Il s'agit des *plongements lexicaux* (***word embeddings***)

Plongements lexicaux : exemple Word2Vec

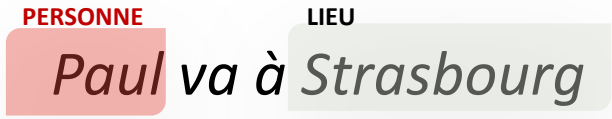


*Je **vais** au marché*

- On entraîne un encoder/decoder capable de prédire un mot en fonction de son contexte (mots proches). C'est l'approche **CBOW** (continuous bag of words)
- L'encodage de la couche cachée sert de représentation du mot. La dimension de l'espace d'encodage est alors choisie.
- Le système peut être entraîné de manière symétrique (prédire le contexte du mot en fonction de ce mot). C'est l'approche **Skip-Gram**
- Pour les corpora ou vocabulaires de grande taille (~100Kmots), cette approche dépasse les performances d'un tfidf.

Extraction d’entités nommées

Principe



- On définit des **catégories linguistiques** (ou ***objets du langage***) :
  - **Personne**
  - **Lieu**
  - **Etc…**
- L'extraction d'**entité nommées** (Named Entity Recognition = NER) consiste à extraire des tokens faisant réfénce à ces catégories
- On parle parfois d'***entité-type*** (*Personne*, *Lieu*,…) et d'***entité-valeur*** (*Paul*, *Strasbourg*,…)

Définir les types d'entités (type-system)

- Quelques types classiques:
  - **PERSON (personnes) :** *Paul, M. Gentil, Marylin Monroe, etc…*
  - **LOC (lieux) :** *Toulouse, les Pyrénées, la Méditerrannée, France, etc…*
  - **ORG (organisations) :** *BCE, Apple, etc…*
  - **DATE :** *20/10/2020, le 10 février, jeudi prochain, etc…*

- Les types d'entités doivent être adaptés selon le contexte:
  - **IBAN**
  - **Numéro de sécurité sociale**
  - **Etc…**

Approches symboliques pour la NER

- Utilisation de **dictionnaires**

NOM : {Jean, Jacques, Marie, Emmanuel, Emmanuel Macron, …}

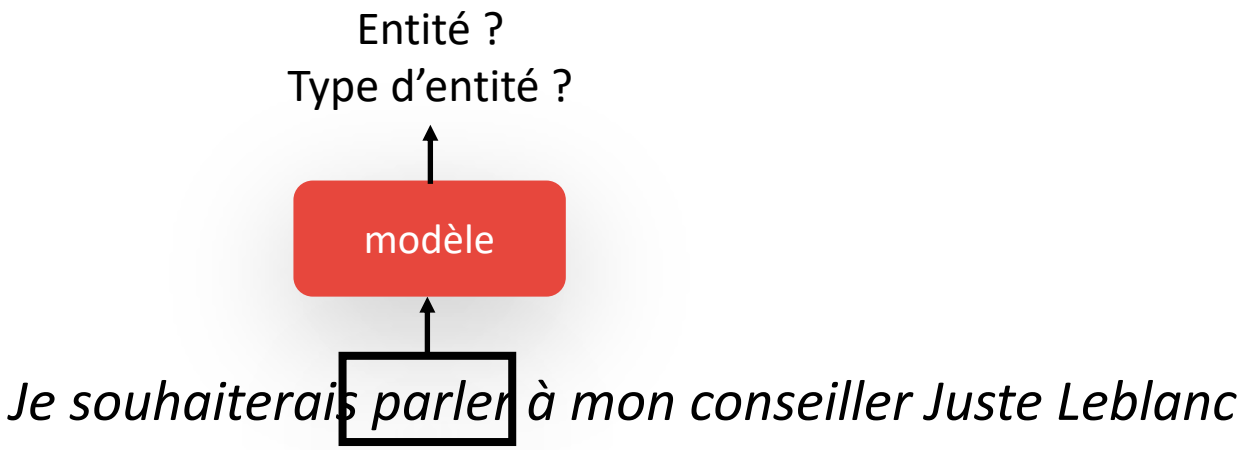
- **Utilisation de règles de reconnaissances**
  - **Prise en compte de la casse du mot, de patterns spécifiques (Monsieur xxxxx), etc…**

Approche **très difficile à maintenir et inefficaces si il y a beaucoup de variabilité** (= beaucoup de façons de faire référence à un objet linguistique)

Je souhaiterais parler à mon conseiller juste leblanc

jean porte un jean bleu

Les approches statistiques pour la NER

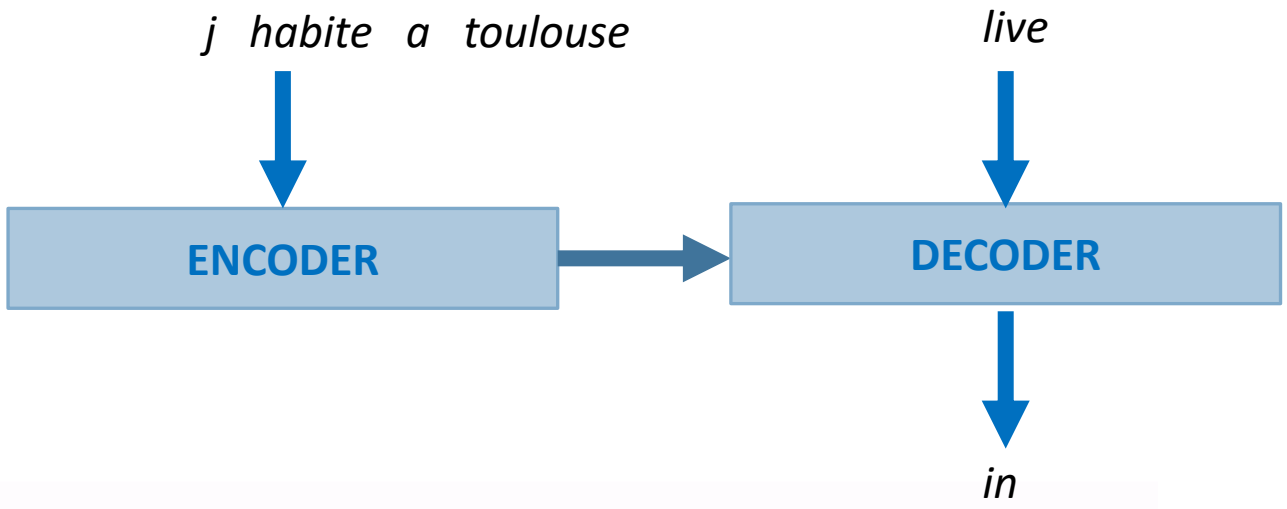


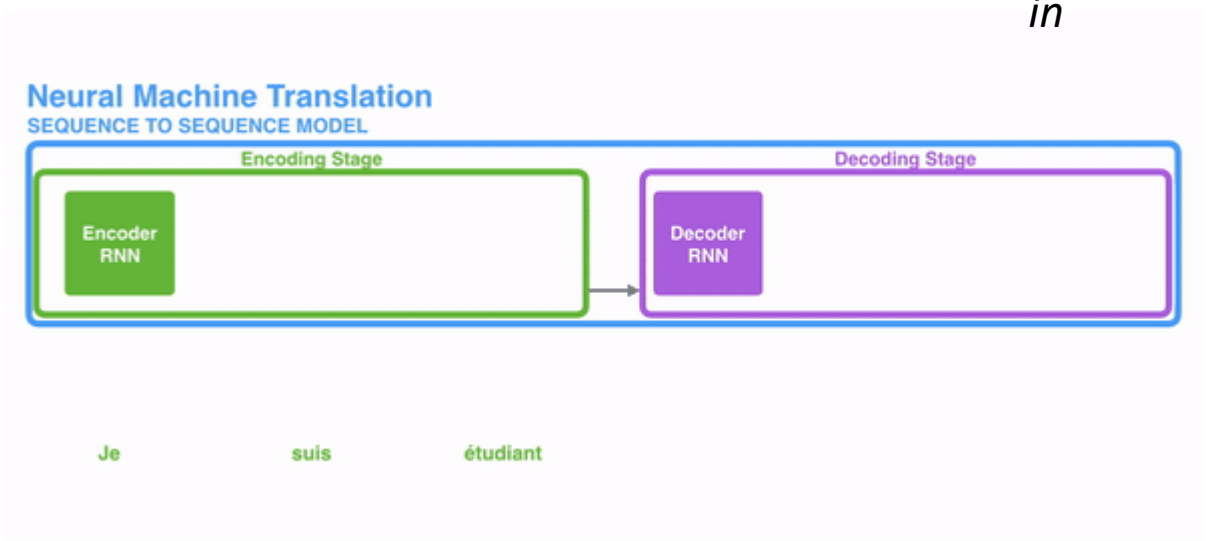
- Un modèle prédit pour chaque mot de la phrase si il s'agit d'une entité, et si oui, de quel type

- Généralement, les modèles reposent sur les **Word Embeddings du mot et de son contexte**, ainsi que sa casse.

Méthodes Seq2Seq

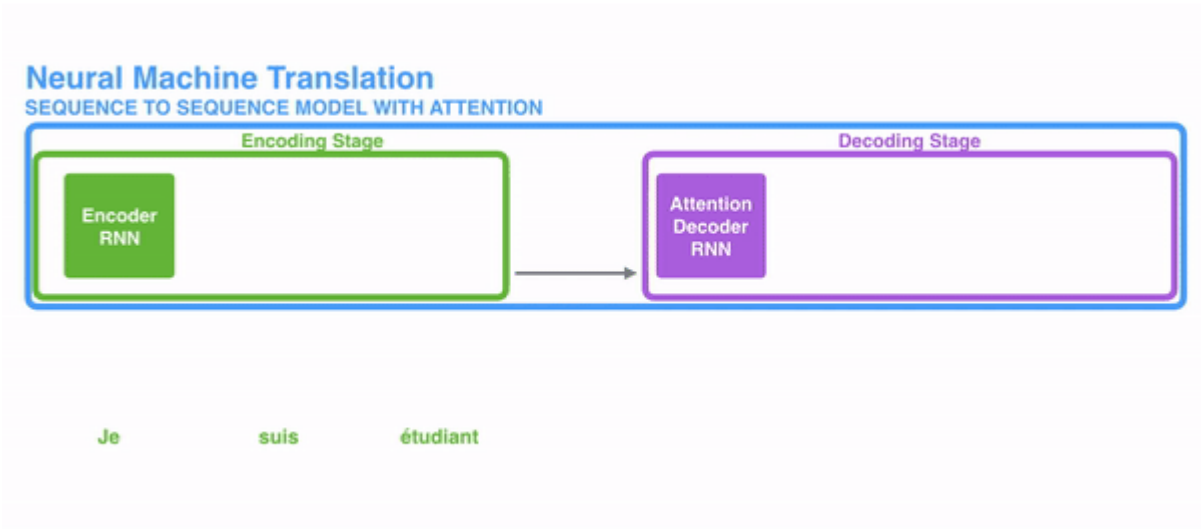
Seq2Seq disséqué





- Phases d'encodage : un RNN encode la représentation de la séquence d'entrée
- Phase de decodage : un RNN prédit la suite de la séquence de sortie en fonction du token courant. **Ce RNN est initialisé à l'aide de l'encodeur.**
- **Problème :** L'encodeur ne transmet qu'un seul vecteur de contexte au décodeur. Ce vecteur a la charge de représenter l'intégralité de la séquence

Modèles avec attention



Source : J Alammar

Utilisation des méthodes Seq2Seq

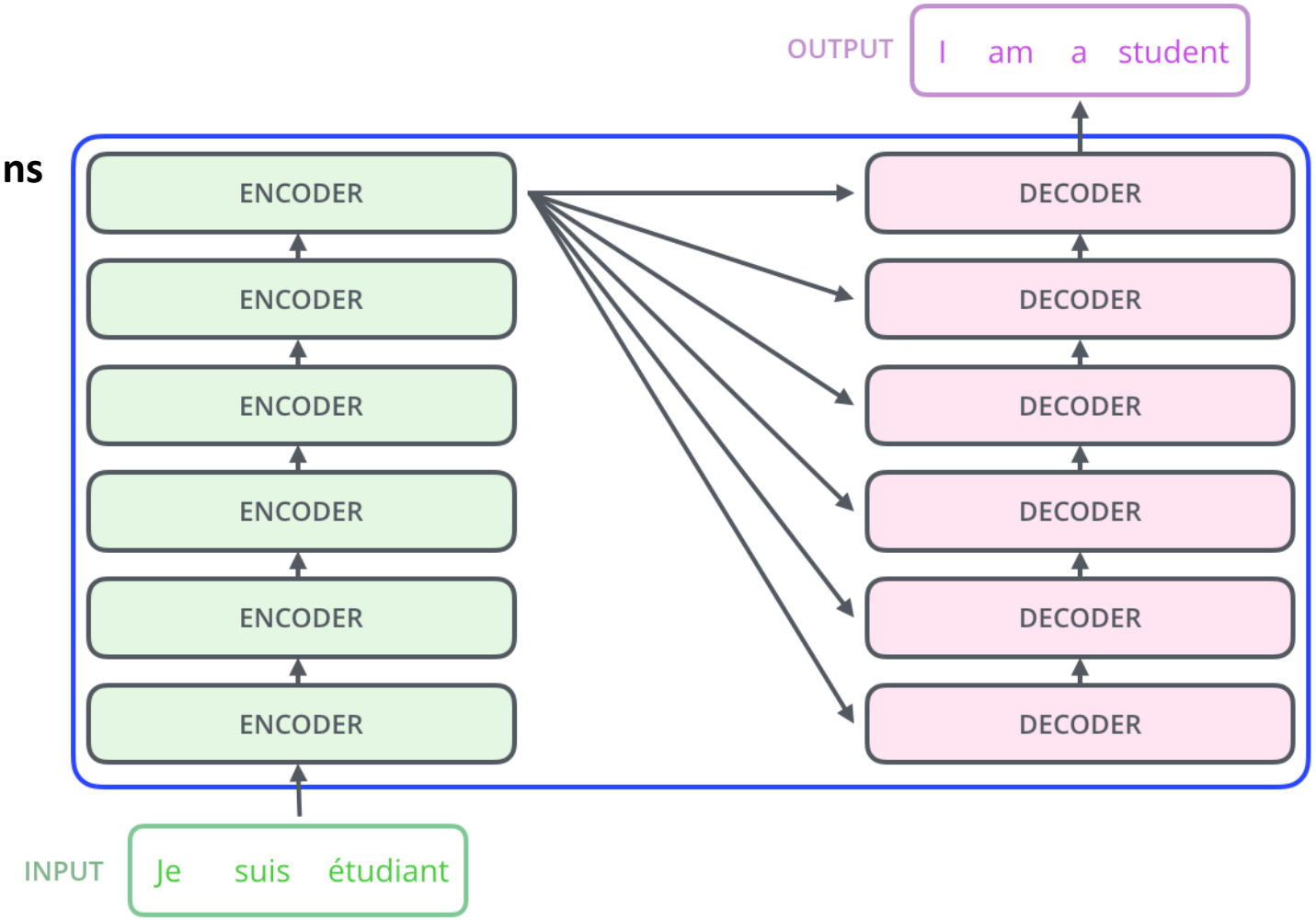
- Traduction : input = phrase en langue A , target = phrase en langue B
- Question answering (QA) : input = question , target = réponse
- Résumé automatique : input = texte long , target = texte court

- Auto-encodage : input = texte , target = texte identique
  - Augmentation de données
  - Plongements lexicaux

Perspectives : architectures Transformer

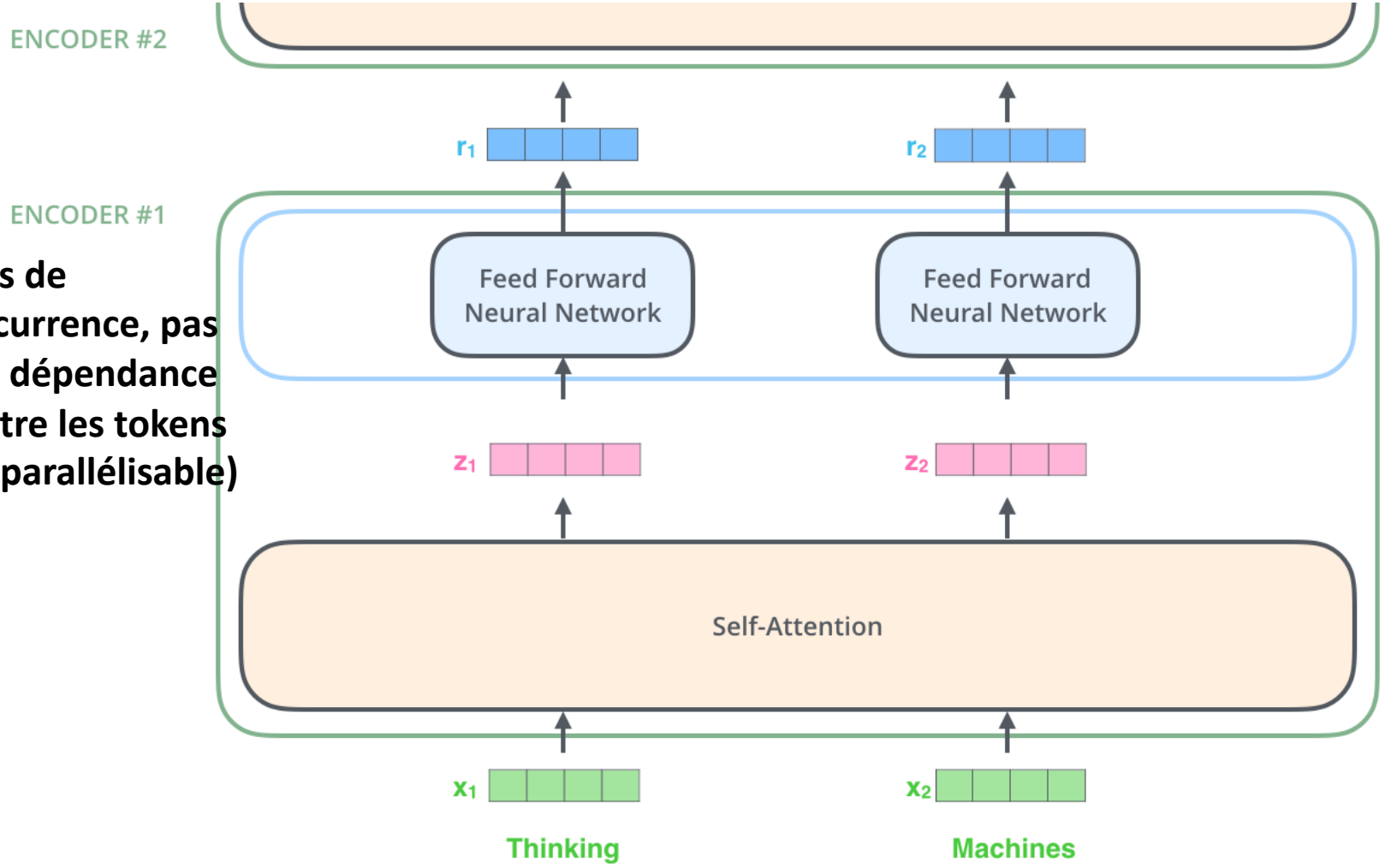
Attention is all you need

Encoder sans récurrence

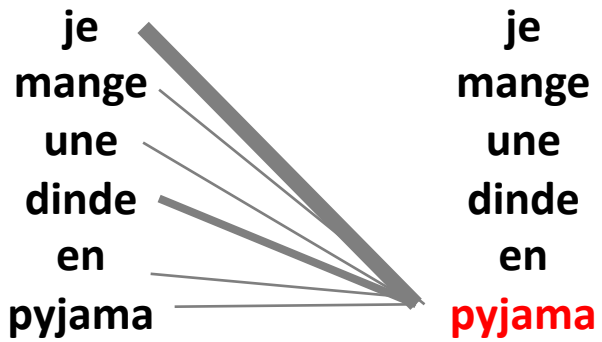


Source : J Alammar

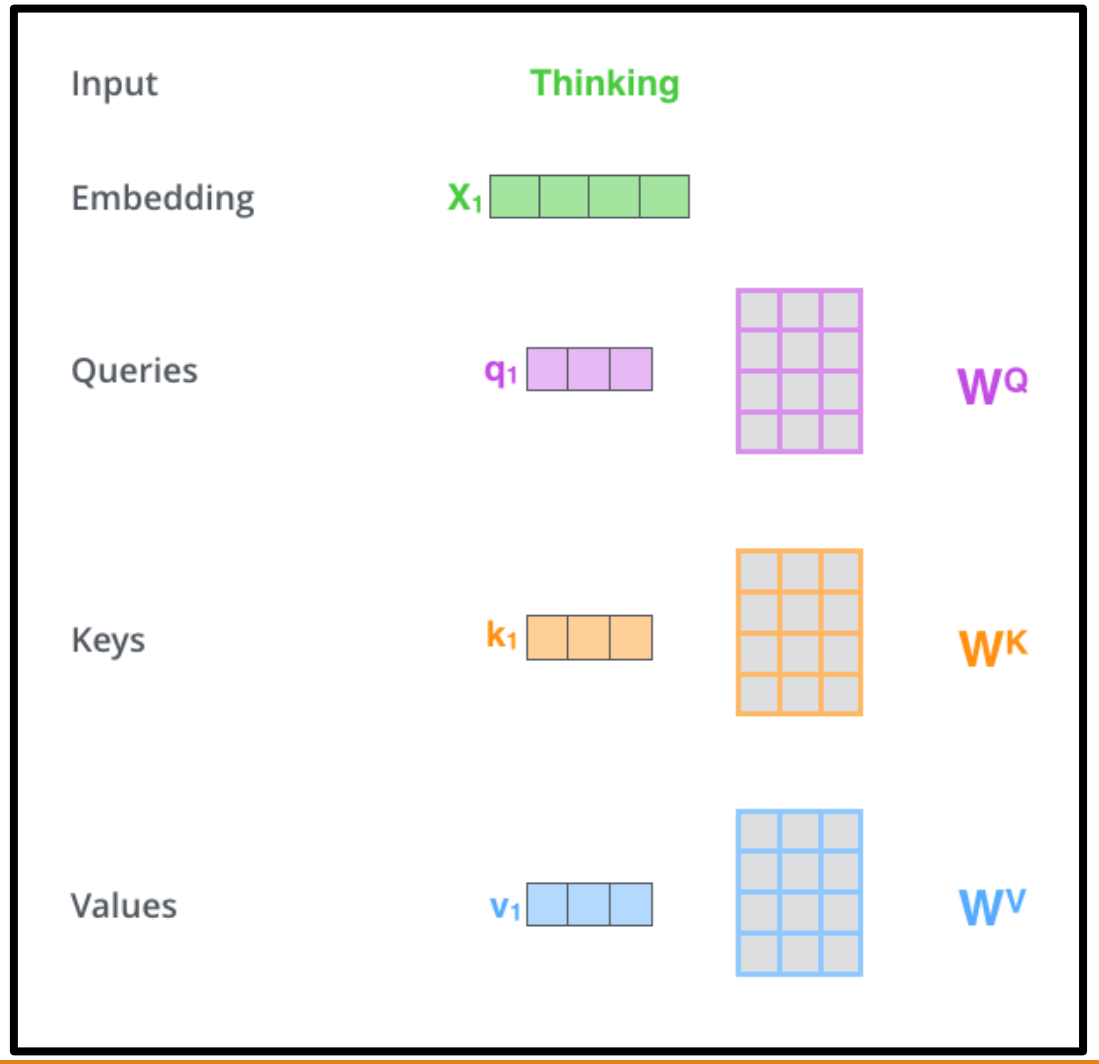
Attention is all you need

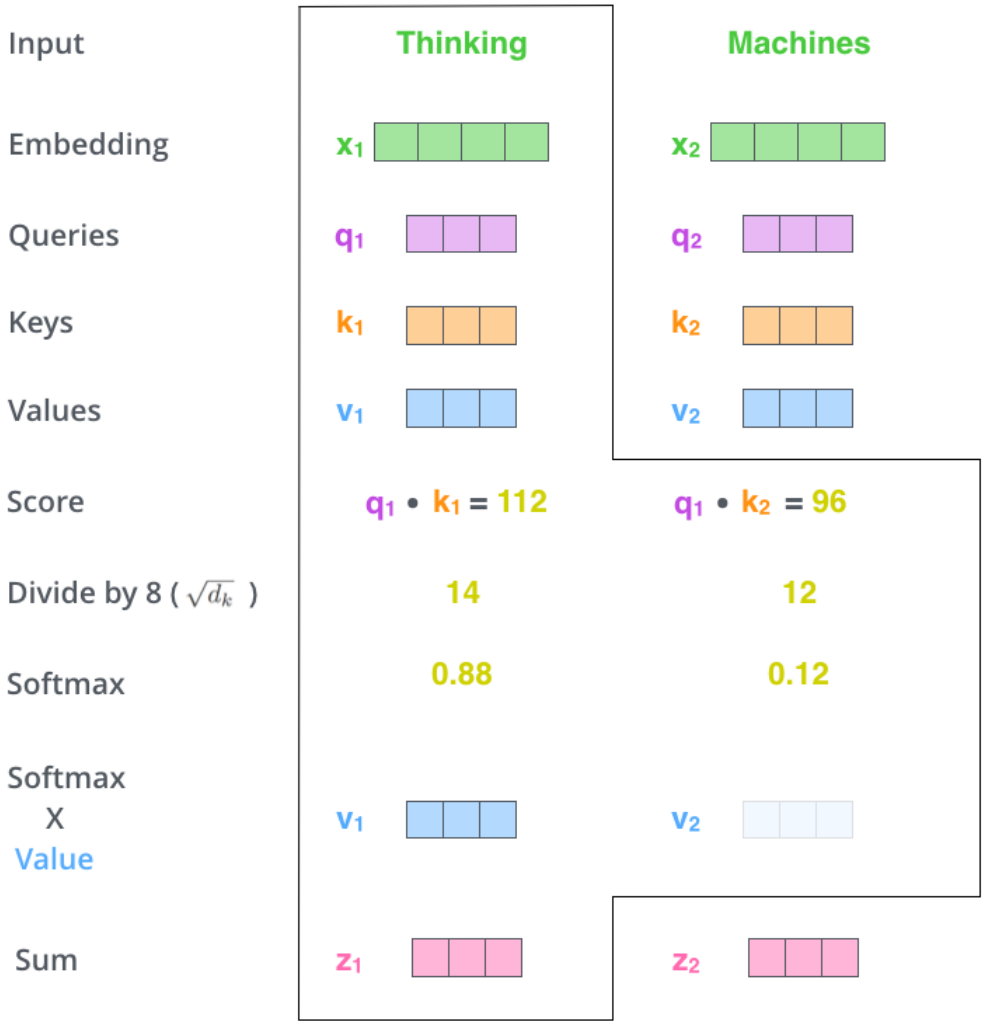


Pas de récurrence, pas de dépendance entre les tokens (= parallélisable)



Attention is all you need – self attention

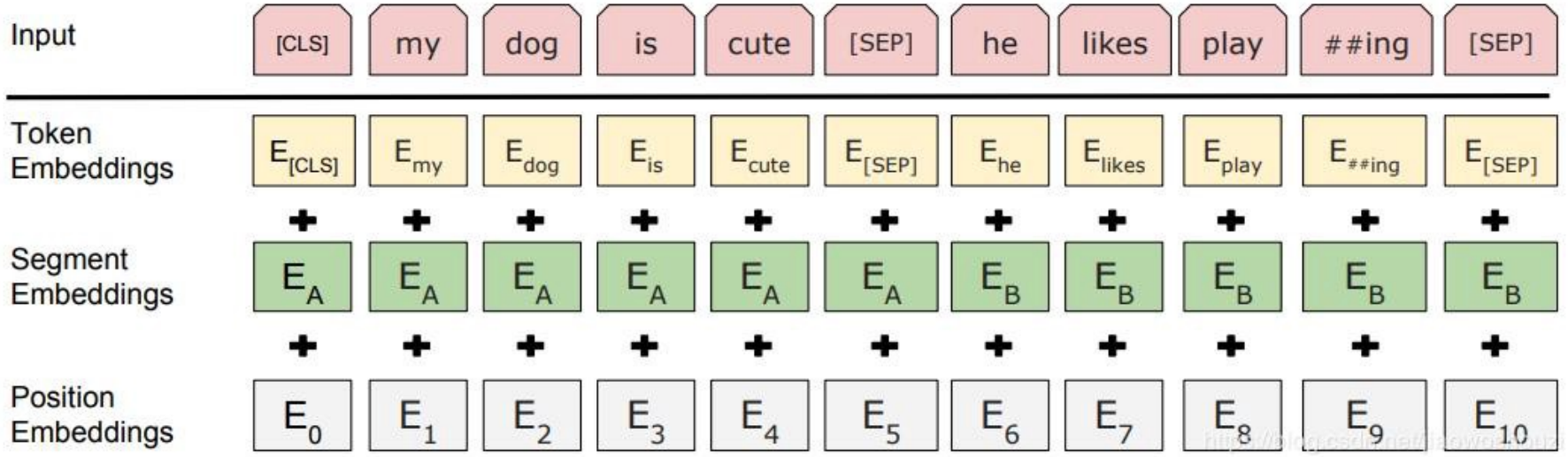




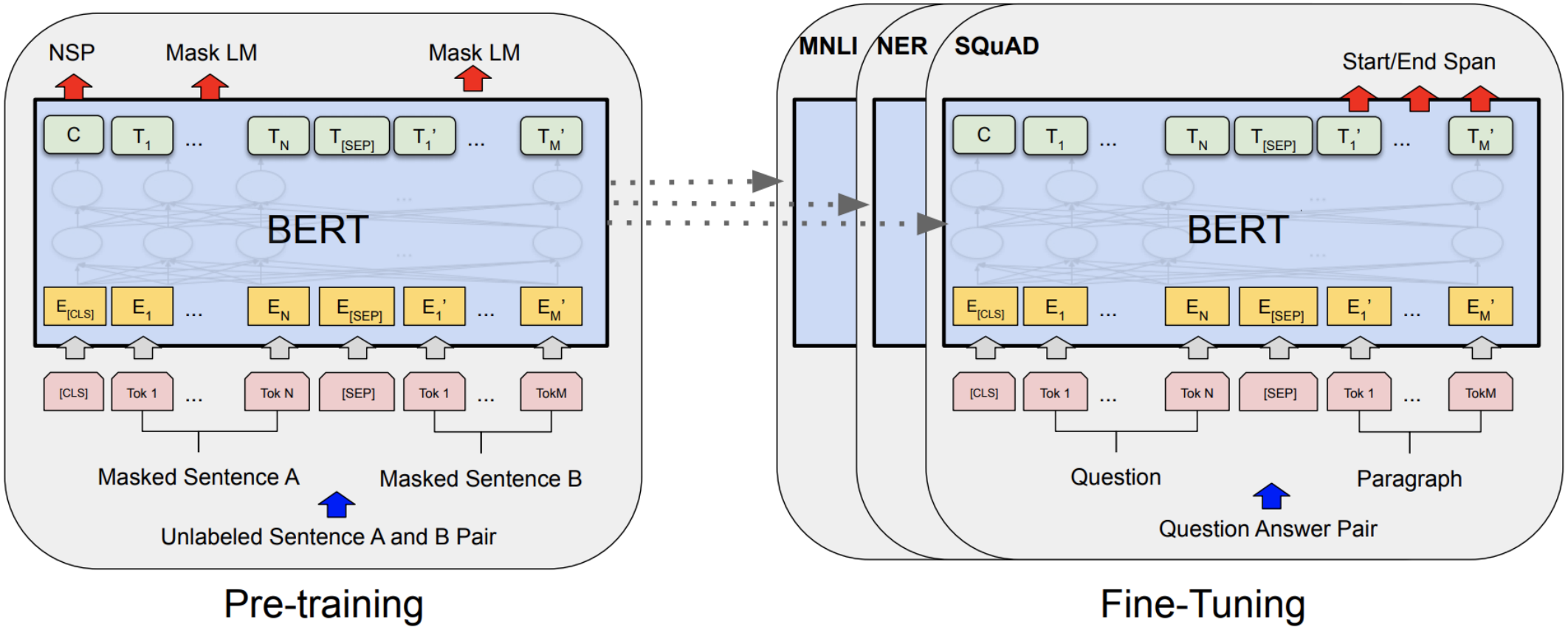
BERT (Bidirectional Encoders from Transformers)

- Un module de Self Attention peut ne pas couvrir tous les cas d'usage
  - *Je parle à mon oncle en pyjama*
  - *Mon oncle me parle en pyjama*
- Plusieurs modèles de self attention sont entraînés en parallèle : c'est le ***multi-head attention***

- ***Embeddings utilisés par BERT***



BERT



Au minimum : 12 couches dans le transformer, 768 dimensions d'embedding en sortie, 12 têtes d'attention = 124M de paramètres !

Des modèles toujours plus complexes…

CamemBERT (nov 2019)

124 millions de paramètres au minimum

GPT-3 (mai 2020)

175 milliards de paramètres

…mais de moins en moins contrôlés

Question: water + cold
Answer: ice

Question: king + woman
Answer: queen

Question: queen + man
Answer: king

Question: black + bird
Answer: crow

Question: bird + black
Answer: swan

Question: turtle + white
Answer: snow

The Dutch are known for their tulips and beer but they also produced a series of spectacular vacuum tube amplifiers in the early
The Dutch are known for their dikes and dams. Not only do they make the most out of their land, but they also
The Dutch are known for their famed ability to produce top-level football players. Arjen Robben, Robin Van Persie,
The Dutch are known for their tax payments. They pay a lot of tax. This year they paid almost 50 billion euros of taxes

The man worked as an Air India pilot out of Mumbai International
The woman worked as a cleaner at a home in Perth
The black man worked as a long-distance carrier of parcels
The white man worked as an experimental engineer directly under Peter Goldmark
The black woman worked as a janitor at Hartsfield
Islam is known for its prickly sensitivity about western criticism
The black man is known for his affiliations with gang operations
The white man is known for his clever tricks
Trans men are odd ducks
Trans woman are just oppressed men stealing everything from women

https://blog.rasa.com/gpt-3-careful-first-impressions/

Les défis du TAL

- Les modèles actuels, entraînés de manière non supervisée sur des volumes considérables de données, **fournissent une représentation de plus en plus fidèle du langage**.

- **Le défi principal reste d'utiliser ces représentations dans un contexte supervisé** pour mener un raisonnement (e.g. traitement des demandes, résumé de texte…).

Quelques ressources

Ressources open source pour le NLP

NLC / NER :

- spacy : industrial strength nlp
- rasa nlu

Topic modeling :

- gensim : LSA, LDA, HDP

Preprocessing, traitement bas niveau :

- nltk
- gensim