Traitement du langage approches séquentielles et génératives, partie II

GAUTIER DURANTIN

GAUTIER.DURANTIN@E-I.COM

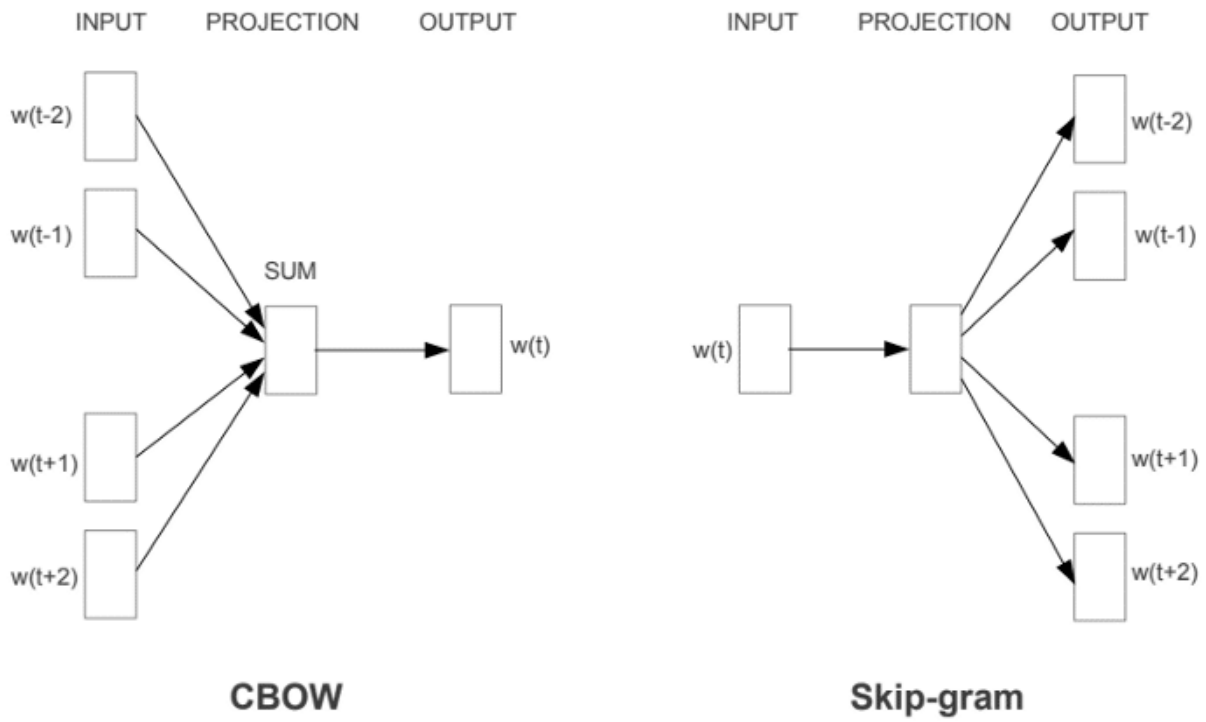
Embeddings classiques

Word2Vec (2012)

- Un réseau à deux couches permet de générer des représentations vectorielles

- On nomme ces représentations Word embeddings ou plongements lexicaux

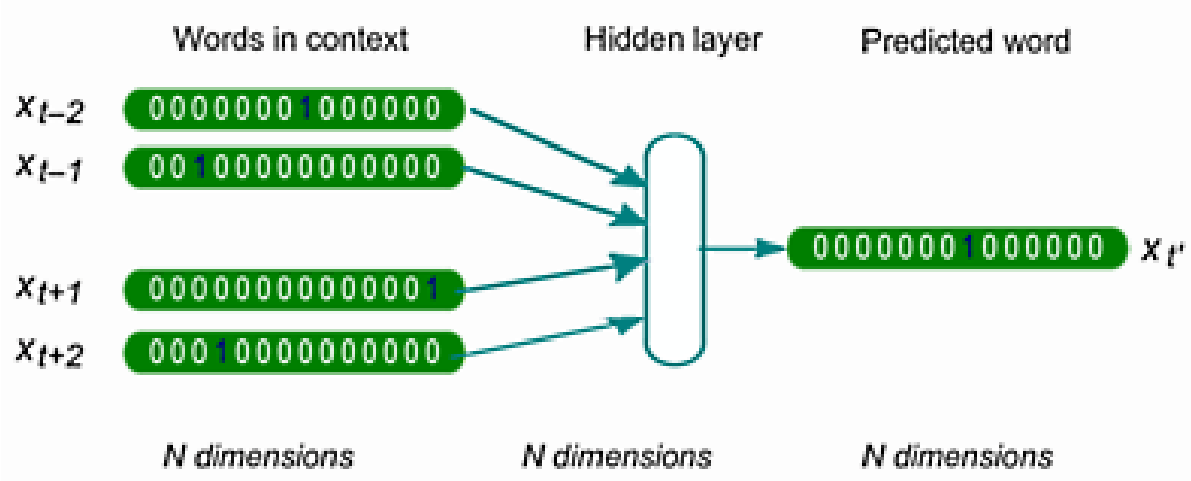
- Deux variantes du Word2Vec existent :
  - CBOW
  - skipgram



Word2Vec - CBOW

- CBOW: Continuous Bag of Words
- On prédit un mot à partir de son contexte

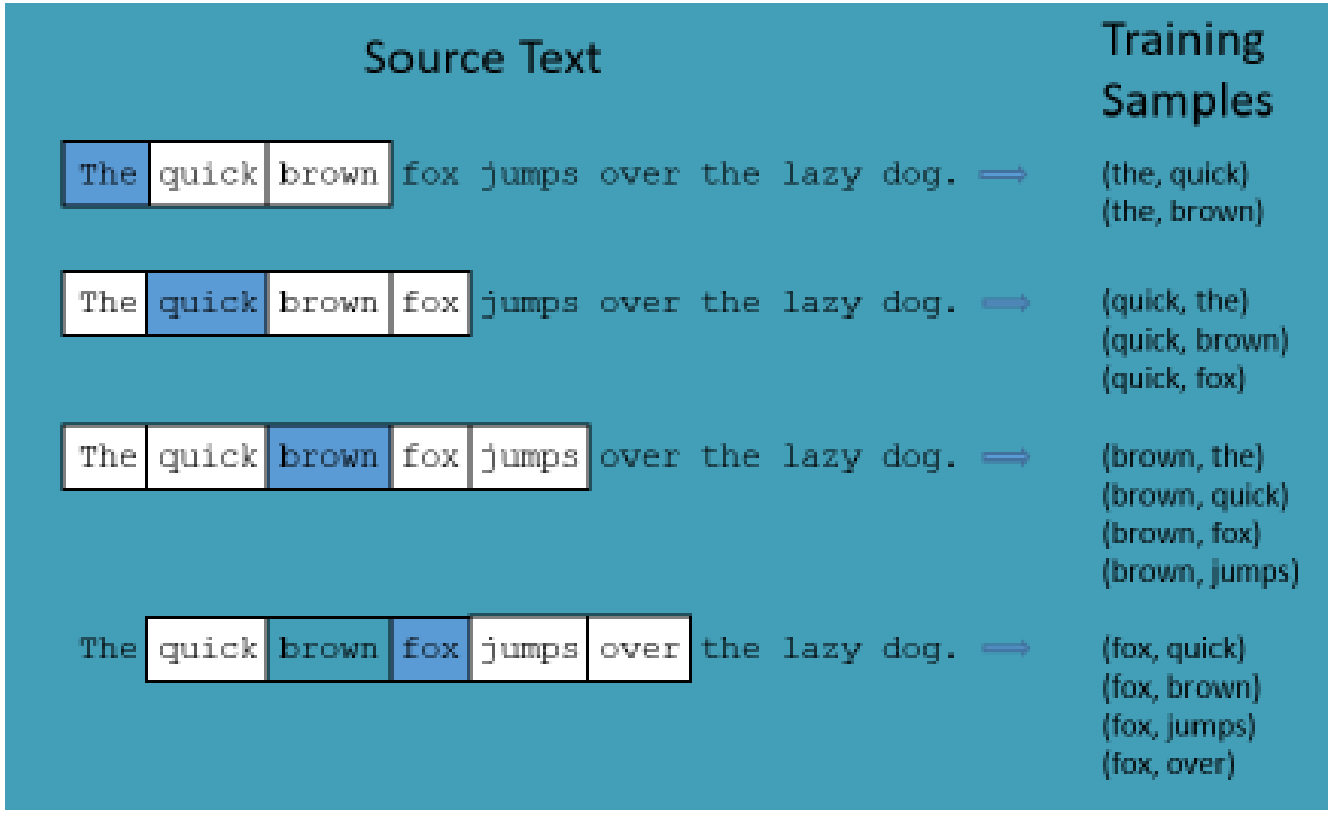
je mange une pomme en dessert



Word2Vec - Skipgram

- On prédit un mot du contexte à partir du token central

je mange une pomme en dessert



Improving Word Representations via Global Context and Multiple Word Prototypes (2012)

- On réalise un clustering sur l'ensemble des contextes afin de maintenir une notion de contexte global

- Un nouveau modèle est entraîné avec une nouvelle représentation du token dans chaque cluster (i.e. si *bank* est proche du cluster 1, il sera tagué bank_1, etc)

- On obtient des prototypes multiples pour un token donné, permettant de maintenir des représentations différentes (gestion de la polysémie)

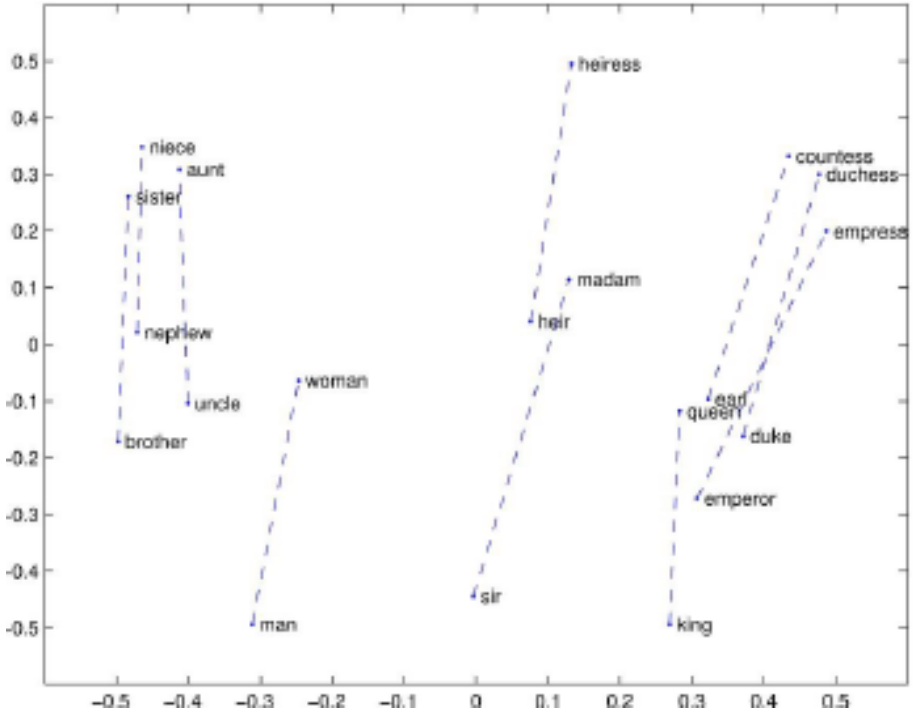
Center Word	Nearest Neighbors
bank_1	corporation, insurance, company
bank_2	shore, coast, direction
star_1	movie, film, radio
star_2	galaxy, planet, moon
cell_1	telephone, smart, phone
cell_2	pathology, molecular, physiology
left_1	close, leave, live
left_2	top, round, right

Table 2: Nearest neighbors of word embeddings learned by our model using the multi-prototype approach based on cosine similarity. The clustering is able to find the different meanings, usages, and parts of speech of the words.

Glove (2014)

- Plutôt que des clusters, on utilise les co-occurrences des tokens pour guider l'apprentissage

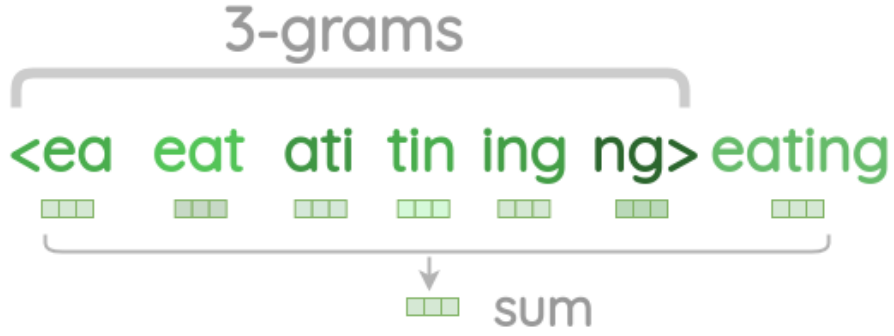
- Cela permet d'avoir une représentation proche du corpus, et de limiter la quantité de données nécessaire pour l'apprentissage



FastText (2016)

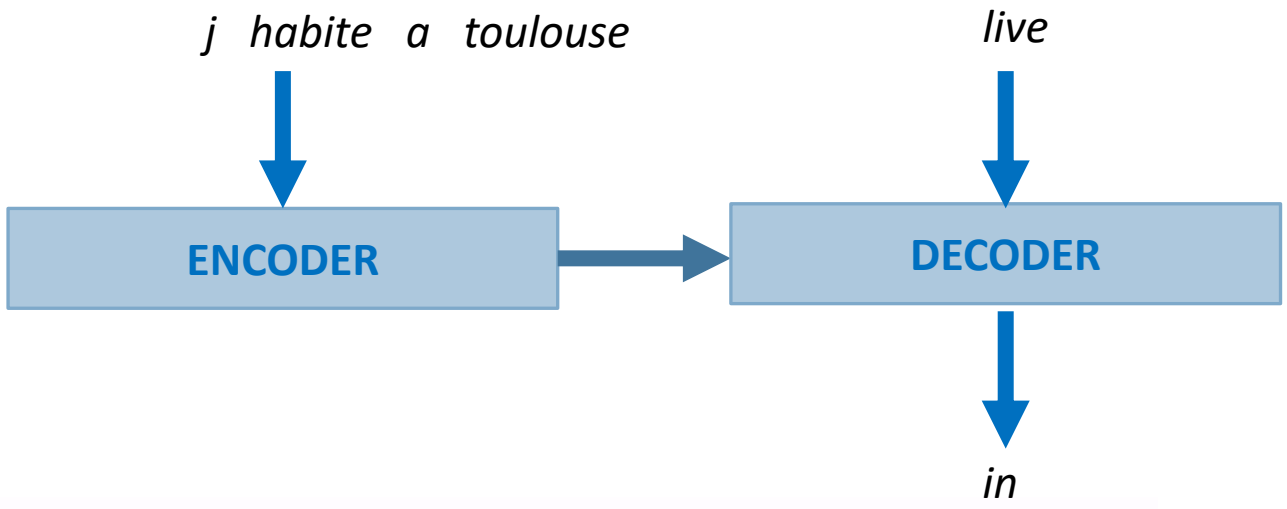
- Utilisation des subwords units

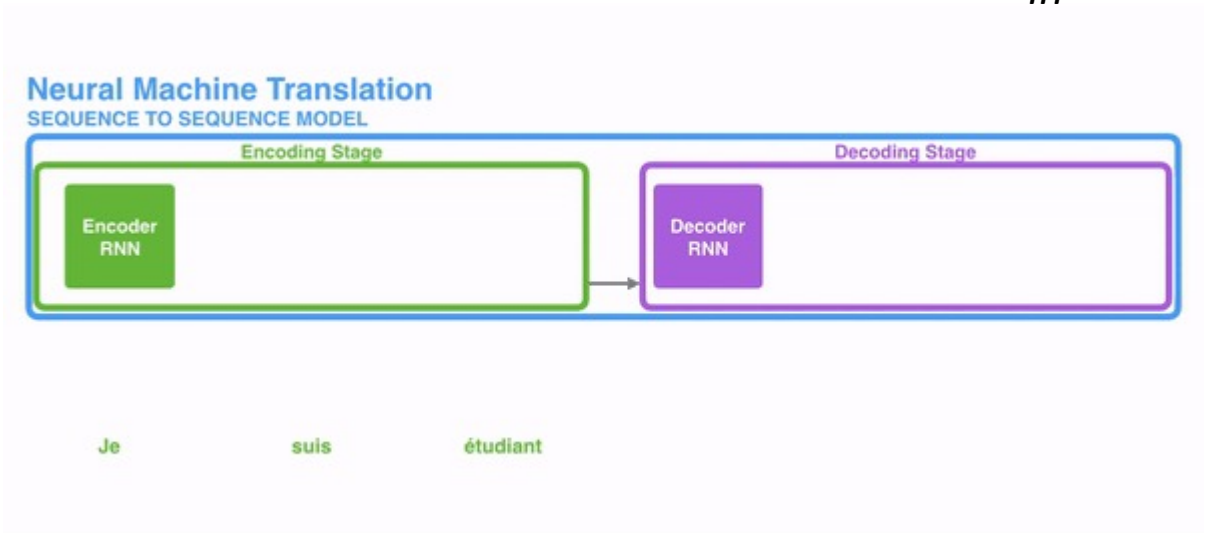
- Cela permet d'avoir une représentation proche pour des mots contenant une faute d'orthographe, ou un mot n'existant pas dans le vocabulaire



Méthodes Seq2Seq

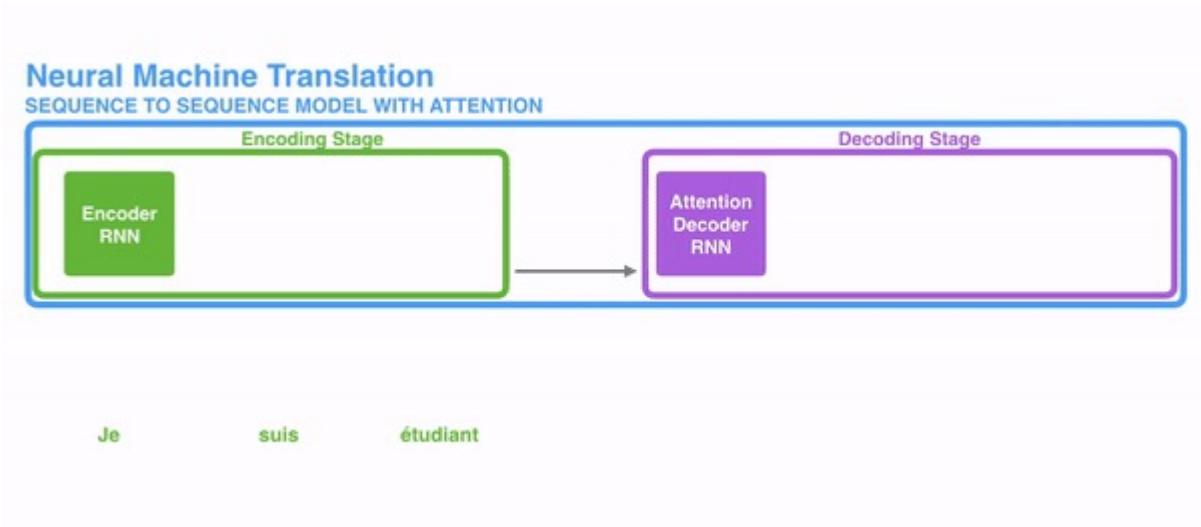
Seq2Seq disséqué





- 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

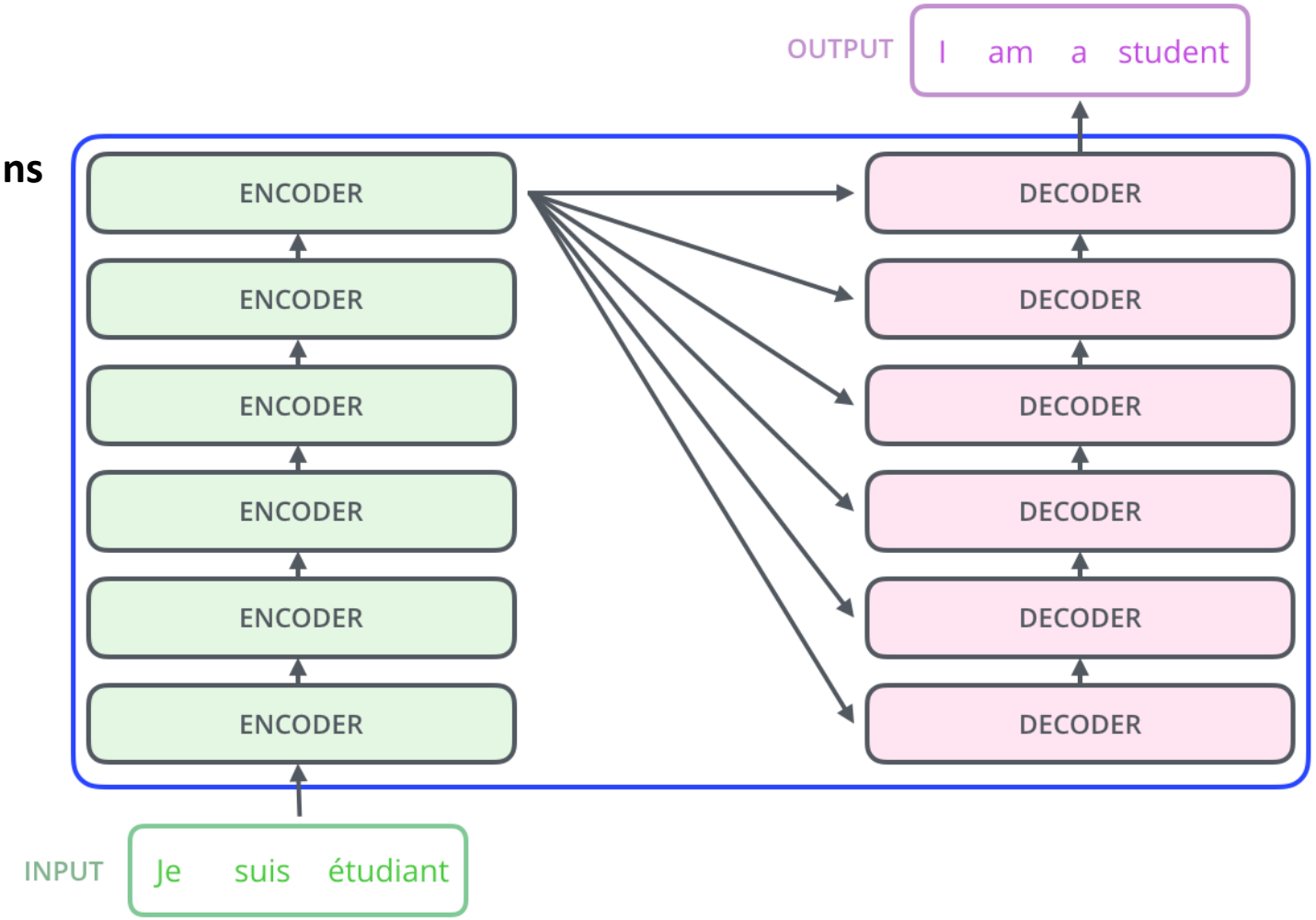
- Traduction : input = phrase en langue A , target = phrase en langue B
- Question answering (QA) : input = question , target = réponse
- Résumé automatique : input = texte long , target = texte court

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

Perspectives : architectures Transformer

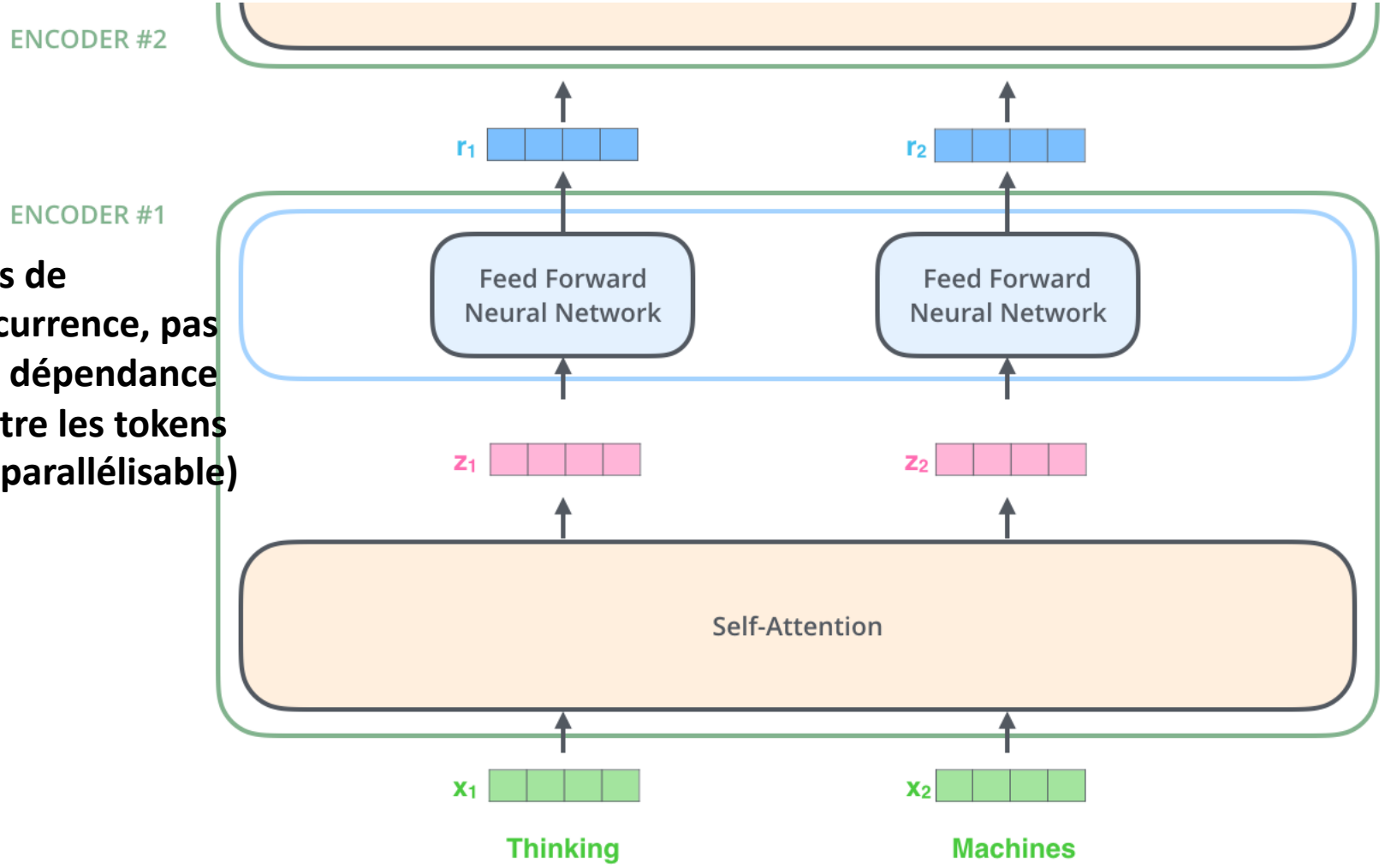
Attention is all you need

Encoder sans récurrence

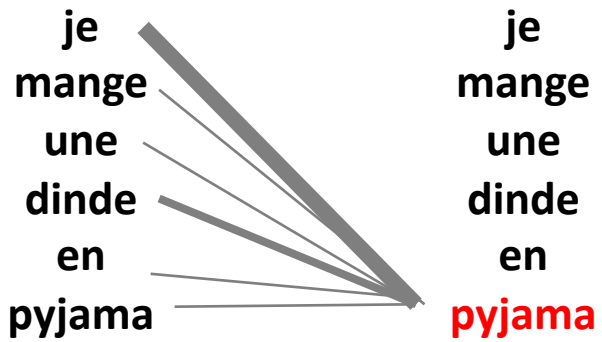


Source : J Alammar

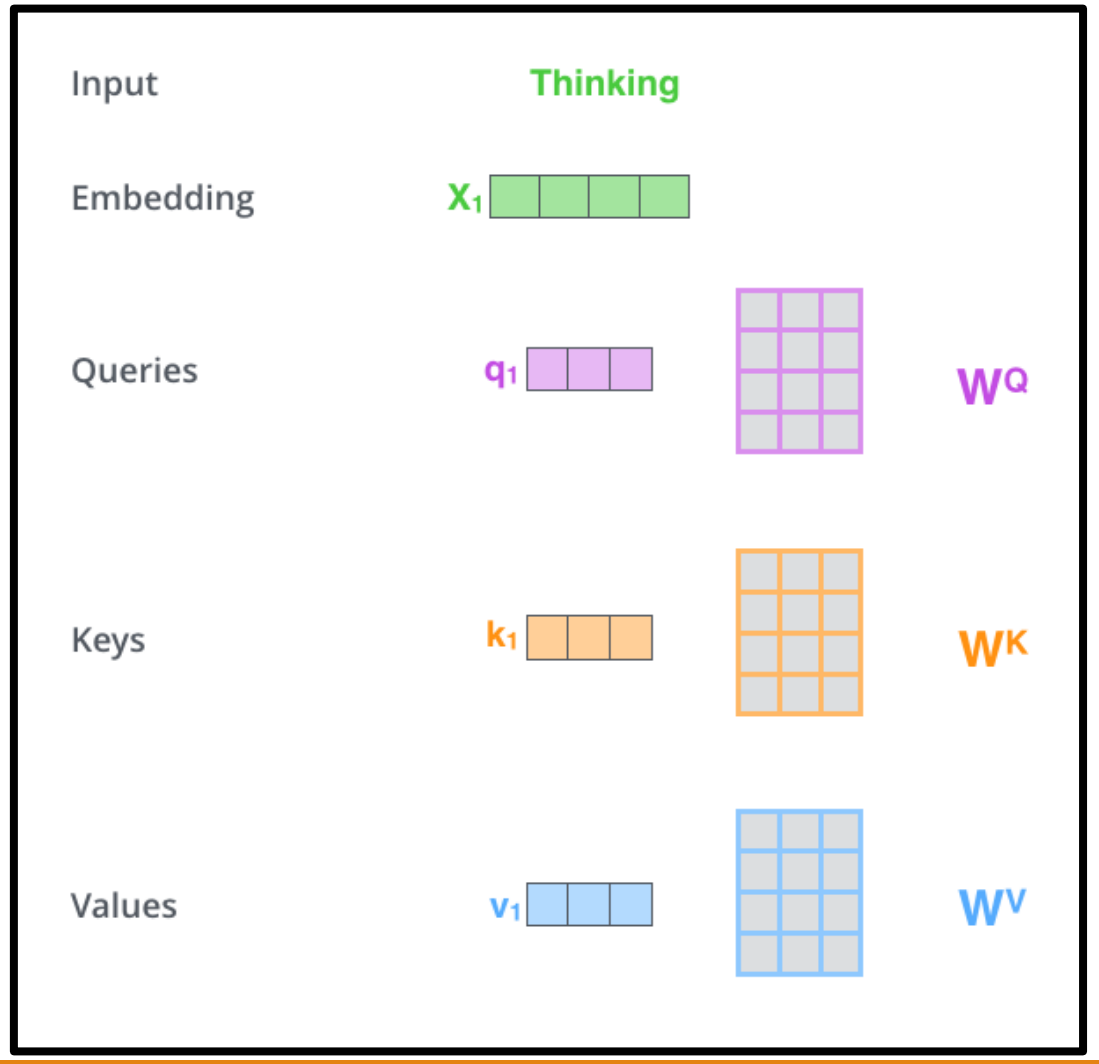
Attention is all you need

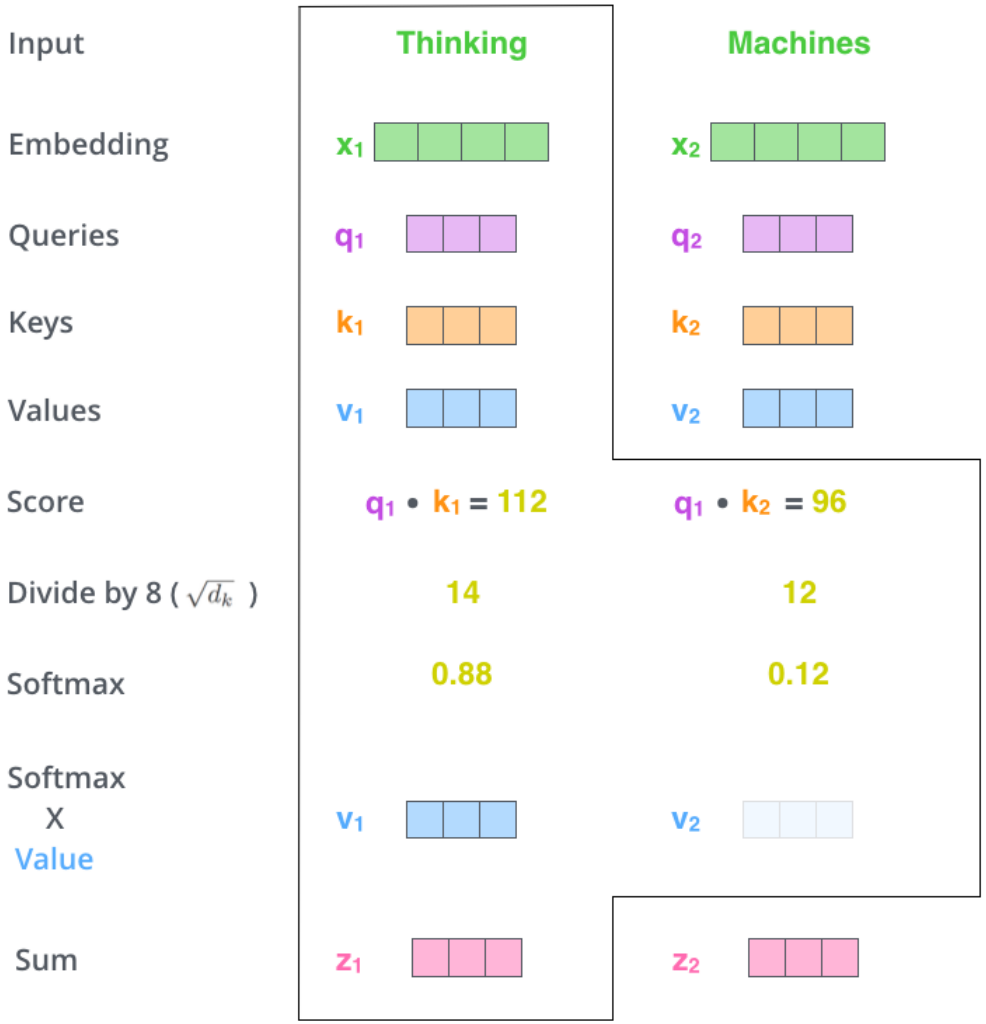


Pas de récurrence, pas de dépendance entre les tokens (= parallélisable)



Attention is all you need – self attention

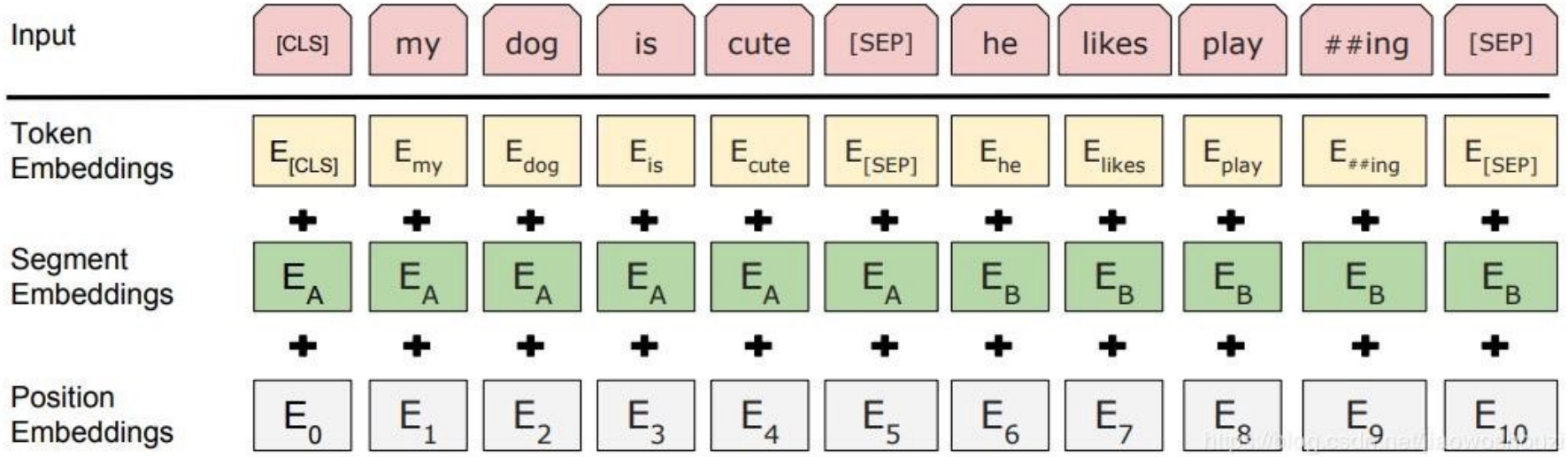




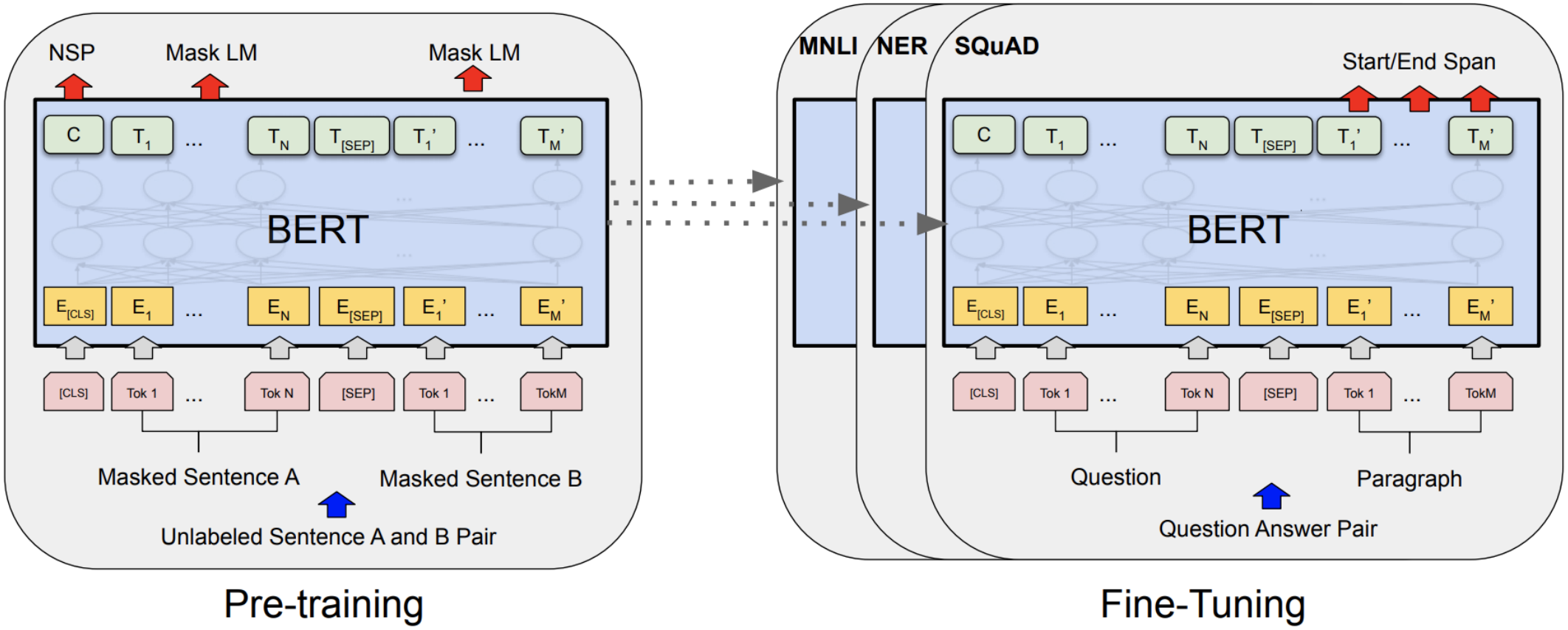
BERT (Bidirectional Encoders from Transformers)

- Un module de Self Attention peut ne pas servir à tous les usages
  - *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 !