Attention et Transformers

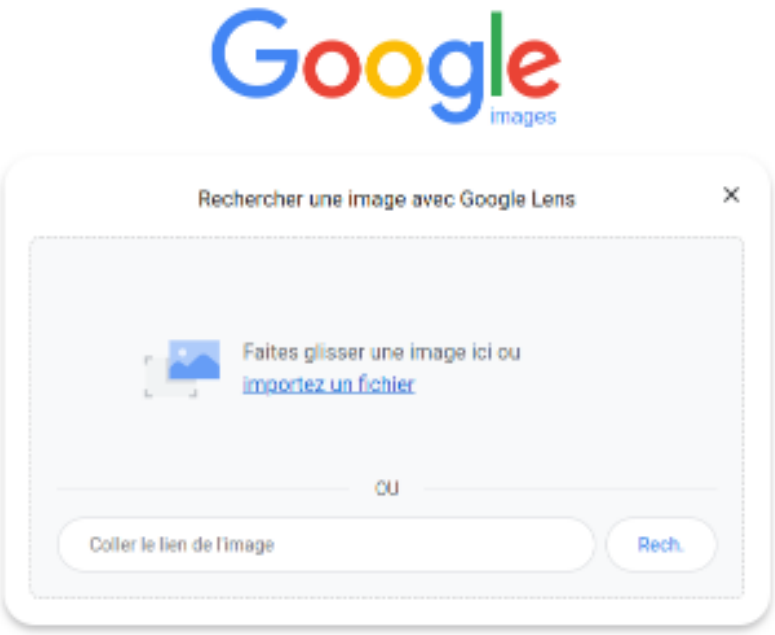
A. Carlier

2023

Plan du cours

1. Recherche d'information
2. Mécanisme d'attention
3. Transformers
4. Applications et évolutions récentes

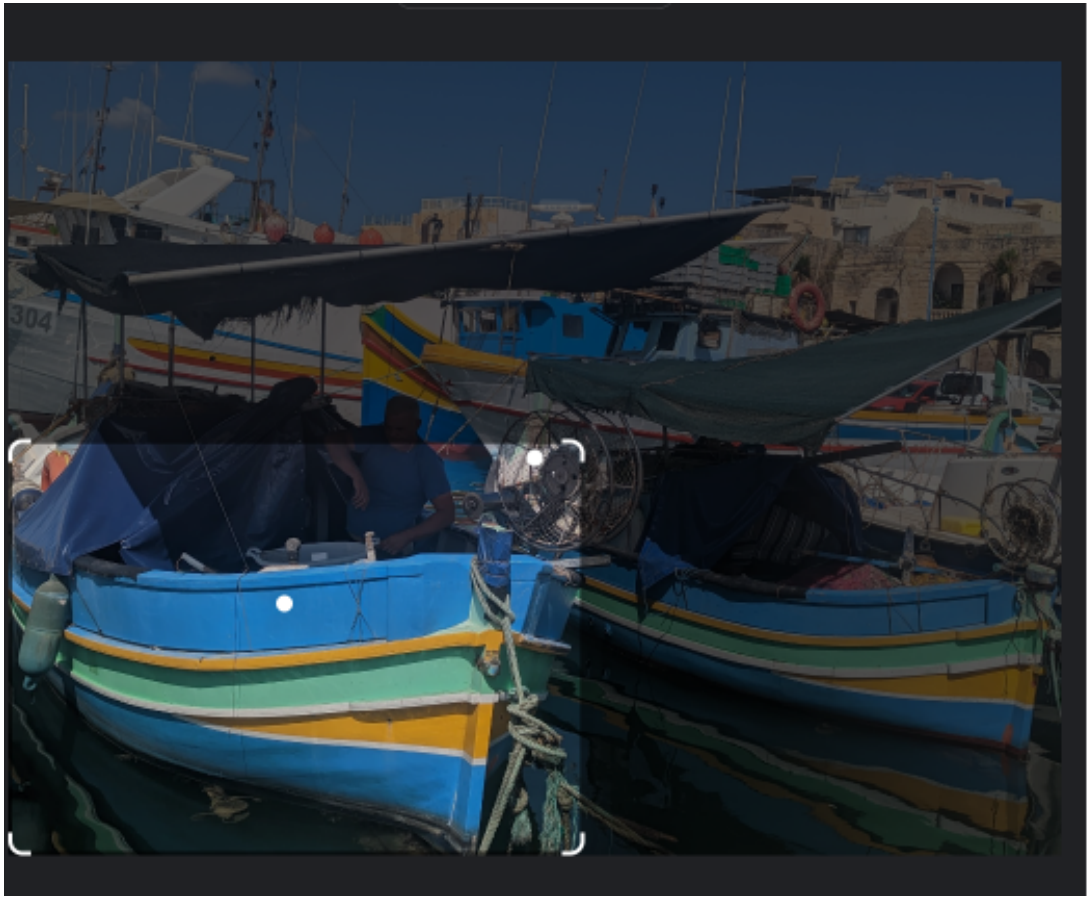
Recherche d'information

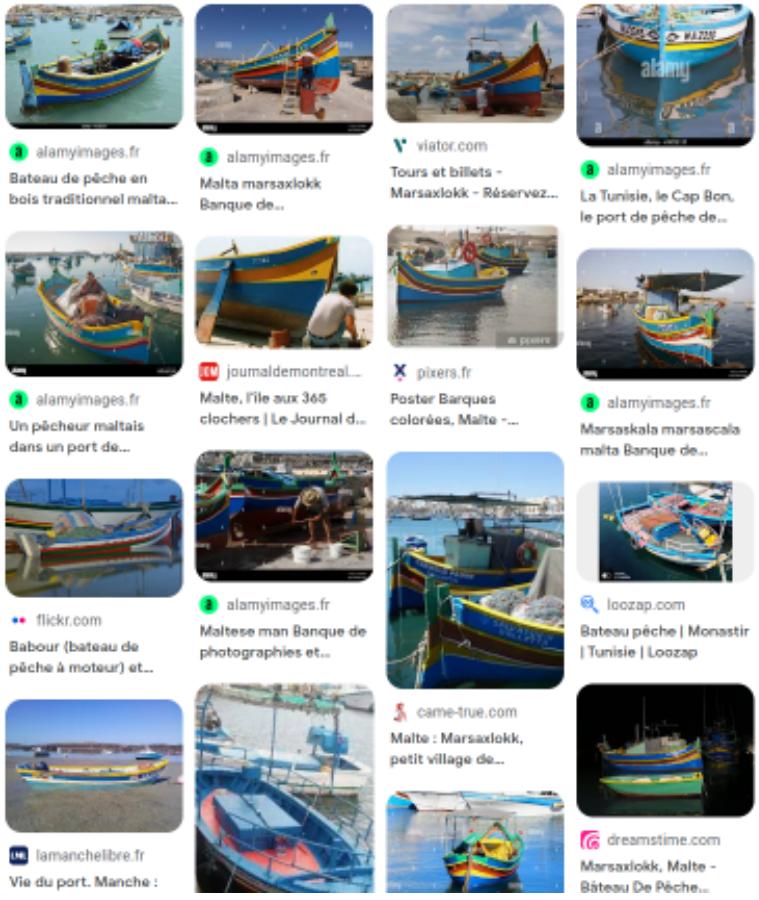


Content-Based Image Retrieval

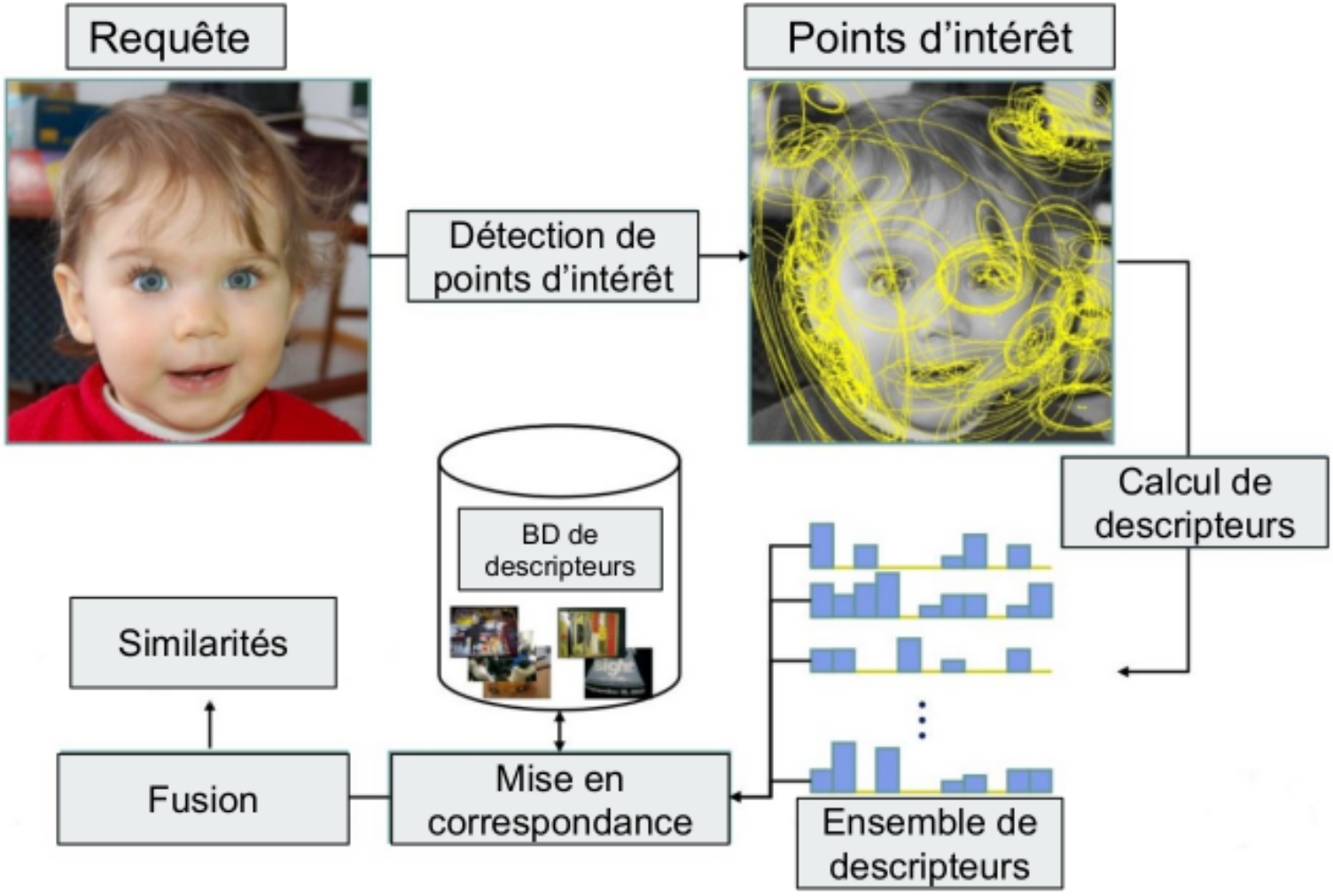
Recherche d'information basée contenu

Recherche d'information

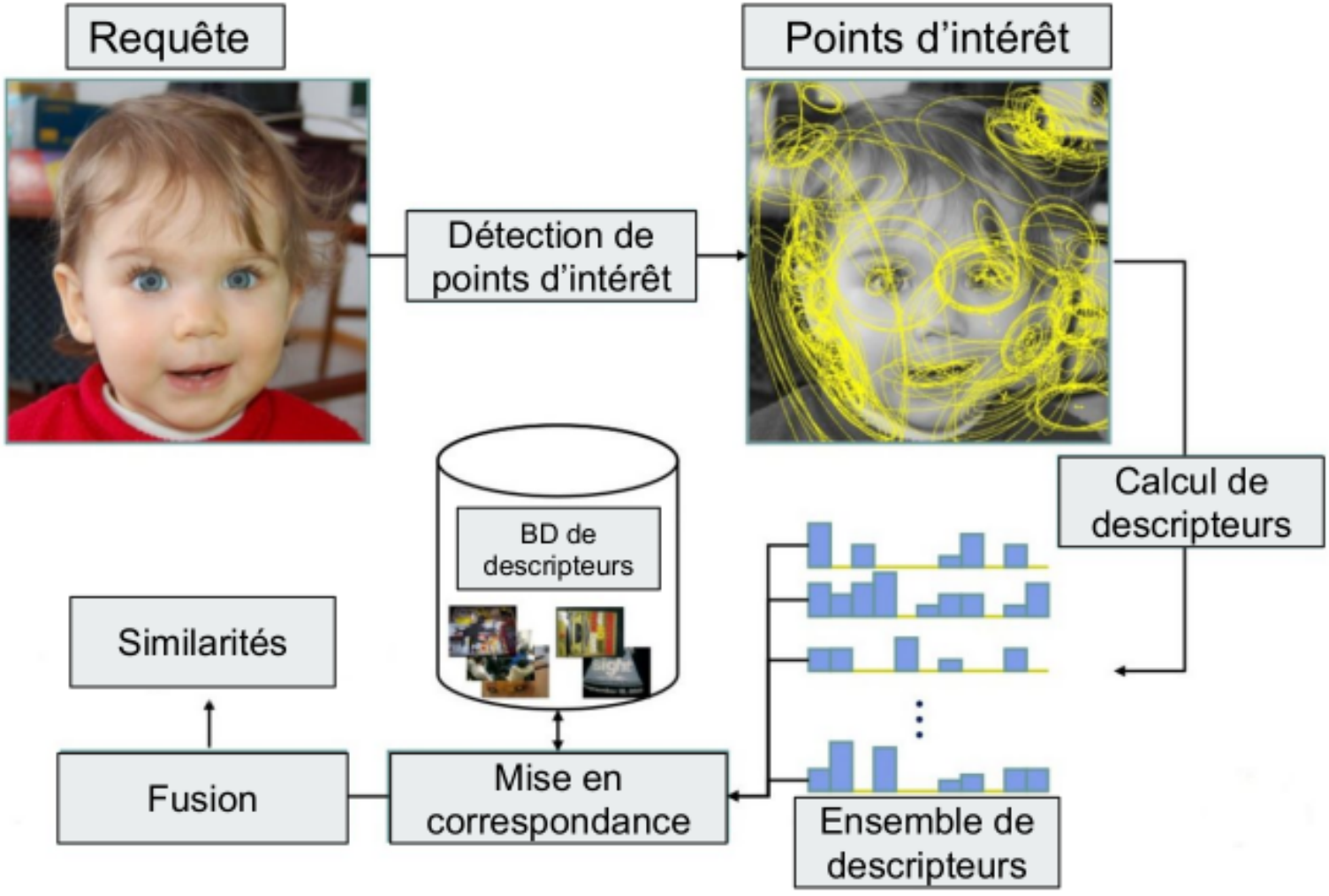




Recherche d'information



Recherche d'information



produit scalaire

Plan du cours

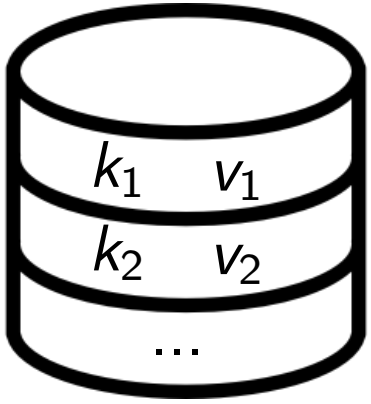
1. Recherche d'information
2. Mécanisme d'attention
3. Transformers
4. Applications et évolutions récentes

Module d'attention

Recherche d'information dans une base de données :

Soit une requête q, on peut comparer la requête aux clés k_i des différentes valeurs stockées en base de données en calculant le produit scalaire qk_i^T.

La réponse renvoyée à cette requête q sera la valeur correspondant à la clé dont le produit scalaire avec q était maximal.

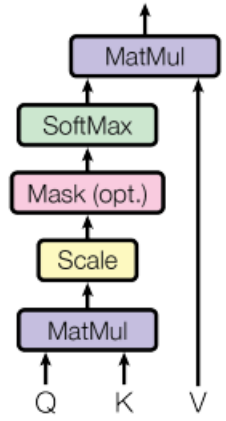


Module d'attention

On appelle **attention** l'implantation de ce mécanisme dans un réseau de neurones. On considère une matrice K contenant les clés, une matrice V contenant les valeurs et une matrice Q contenant les requêtes :

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

Les clés et les requêtes sont de même dimension d_k et le facteur de normalisation permet de conserver des valeurs dans une zone où les gradients ne sont pas trop faibles.

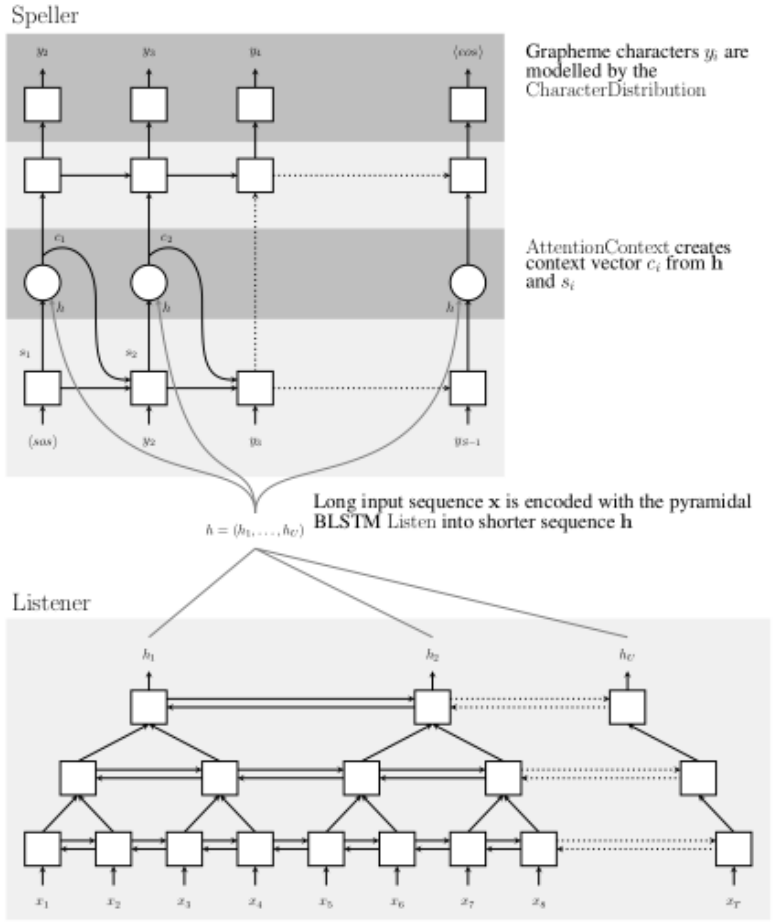


Listen, Attend and Spell

En 2015, l'état de l'art des modèles *Seq2seq* (séquence à séquence) utilise des LSTM et un modèle d'attention.

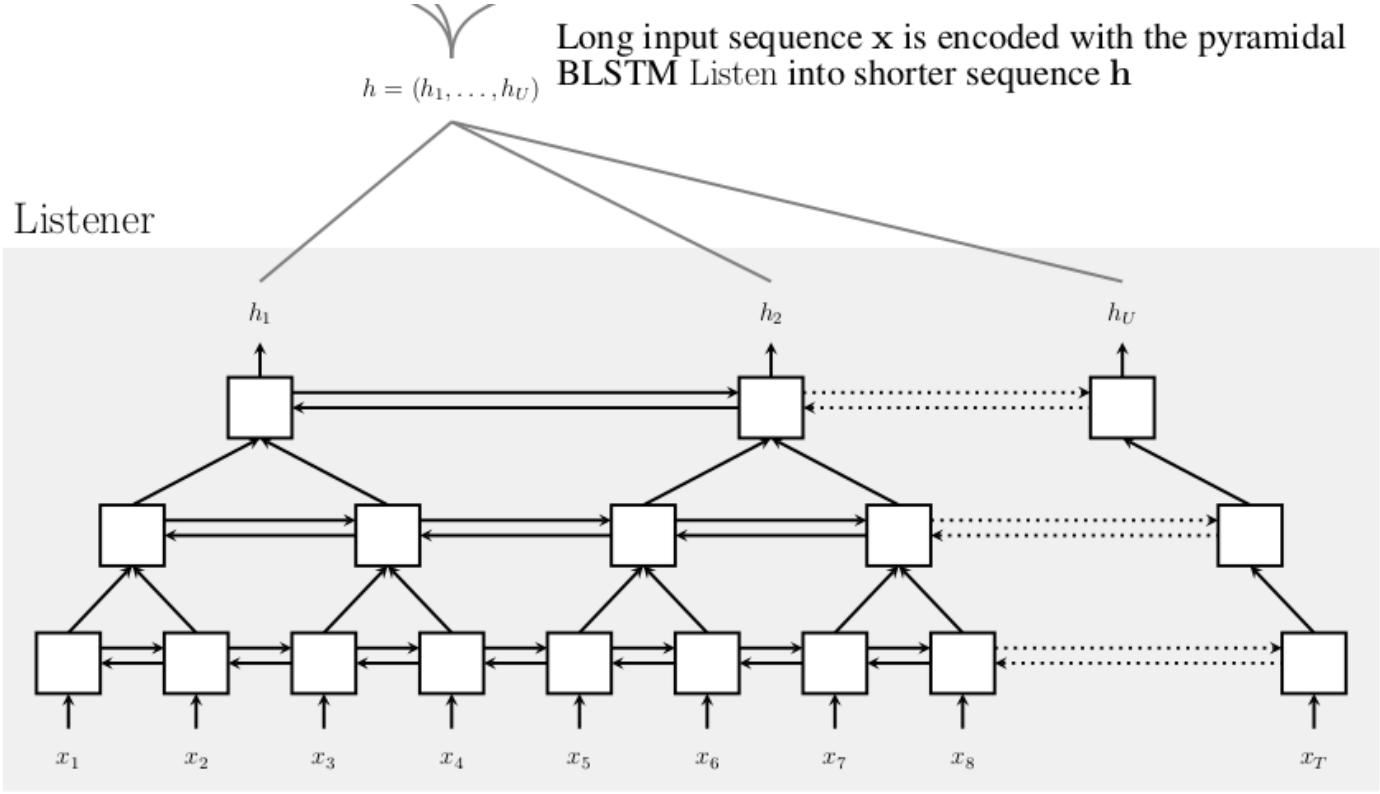
Il s'agit d'un modèle encodeur-décodeur pour la transcription automatique, où l'encodeur "écoute" (Listener) le signal audio et le décodeur "épelle" (spell) sa transcription.

[Chan 2015] Listen, Attend and Spell



Listen

LSTM bidirectionnel pyramidal, avec diminution de la dimension à chaque nouvelle couche pour simplifier le signal d'entrée.



[Chan 2015] Listen, Attend and Spell

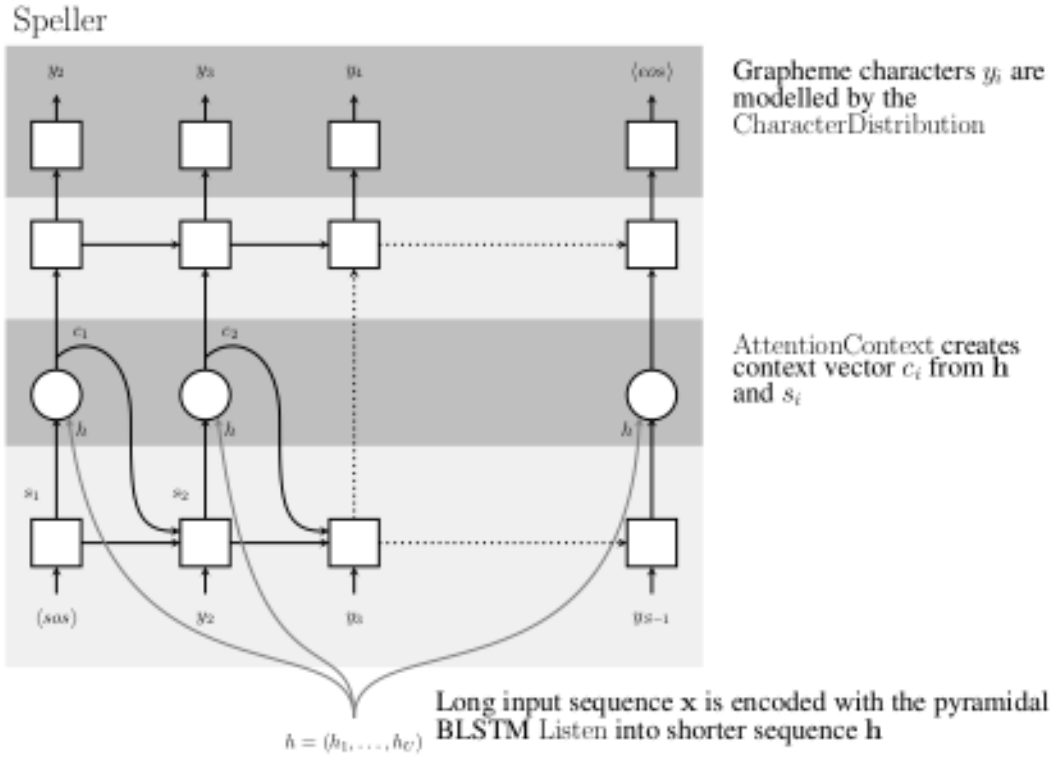
Spell

LSTM avec attention qui permet de générer une séquence en portant, à différents moments, un poids (une attention) particulier à différents éléments de la séquence *h*.

$$e_{i,u} = \langle \phi(s_i), \psi(h_u) \rangle$$

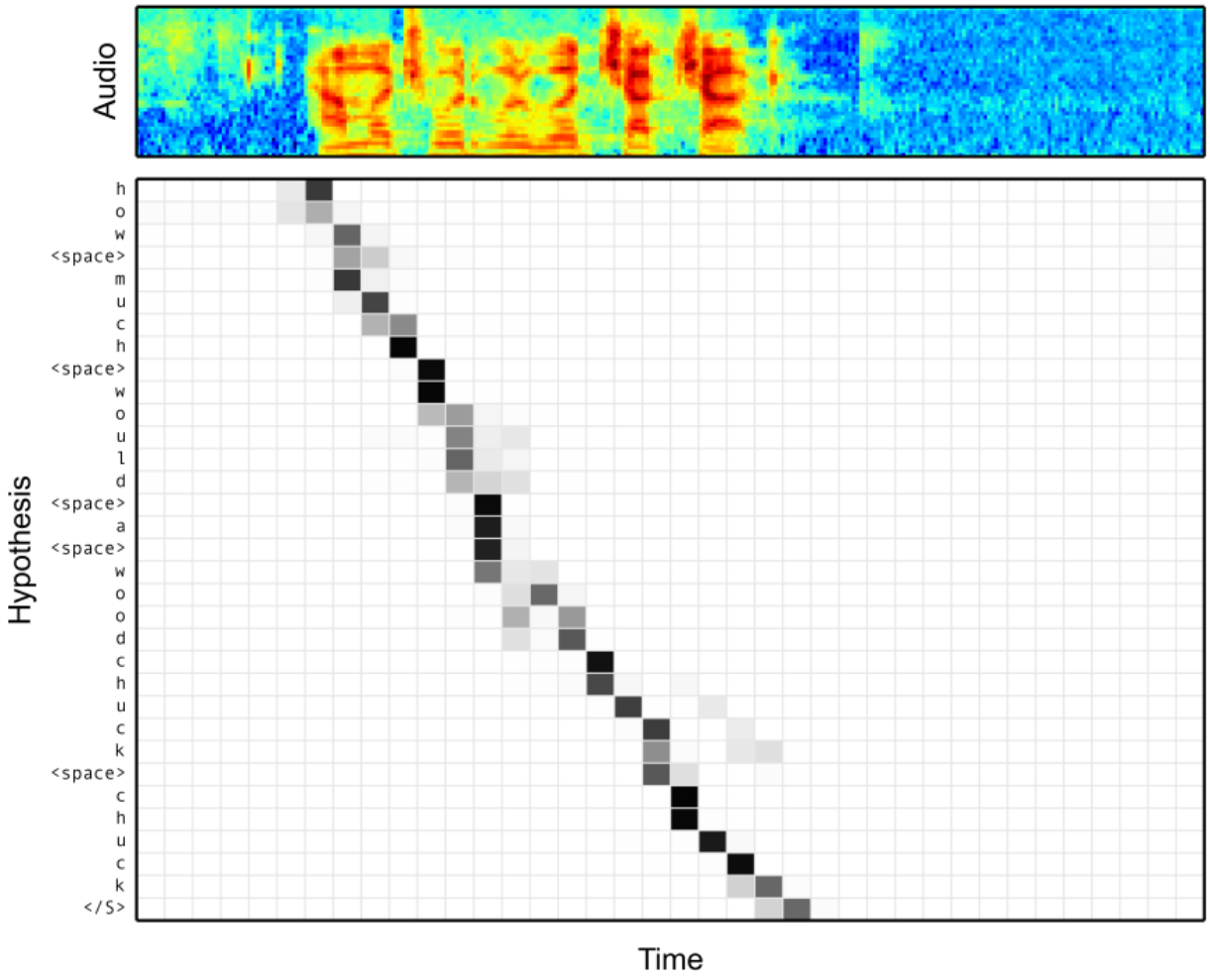
$$\alpha_{i,u} = \frac{\exp(e_{i,u})}{\sum_u \exp(e_{i,u})}$$

$$c_i = \sum_u \alpha_{i,u} h_u$$



[Chan 2015] Listen, Attend and Spell

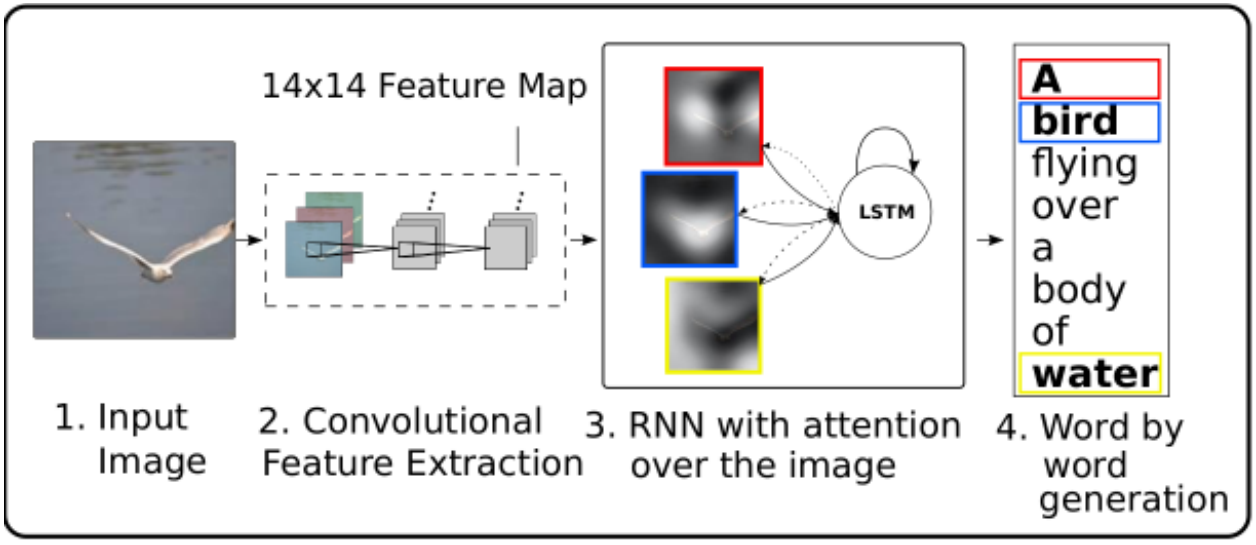
Listen, Attend and Spell

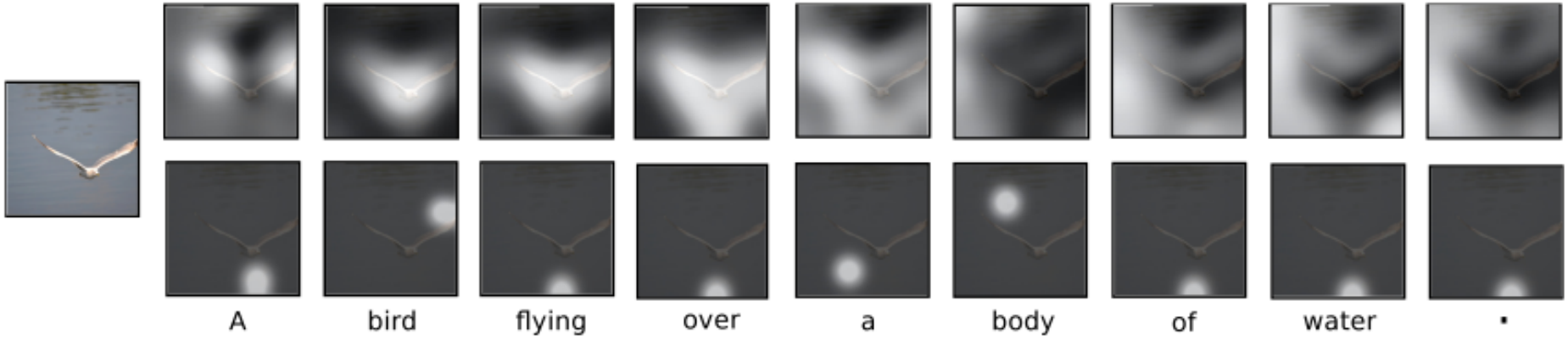


Visualisation de l'attention portée au signal d'entrée pour la génération de chaque caractère en sortie

[Chan 2015] Listen, Attend and Spell

Show, Attend and Tell





[Xu 2016] Show, Attend and Tell : Neural Image Caption Generation with Visual Attention

Show, Attend and Tell

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

[Xu 2016] Show, Attend and Tell : Neural Image Caption Generation with Visual Attention

Plan du cours

1. Recherche d'information
2. Mécanisme d'attention
3. Transformers
4. Applications et évolutions récentes

L'article fondateur

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

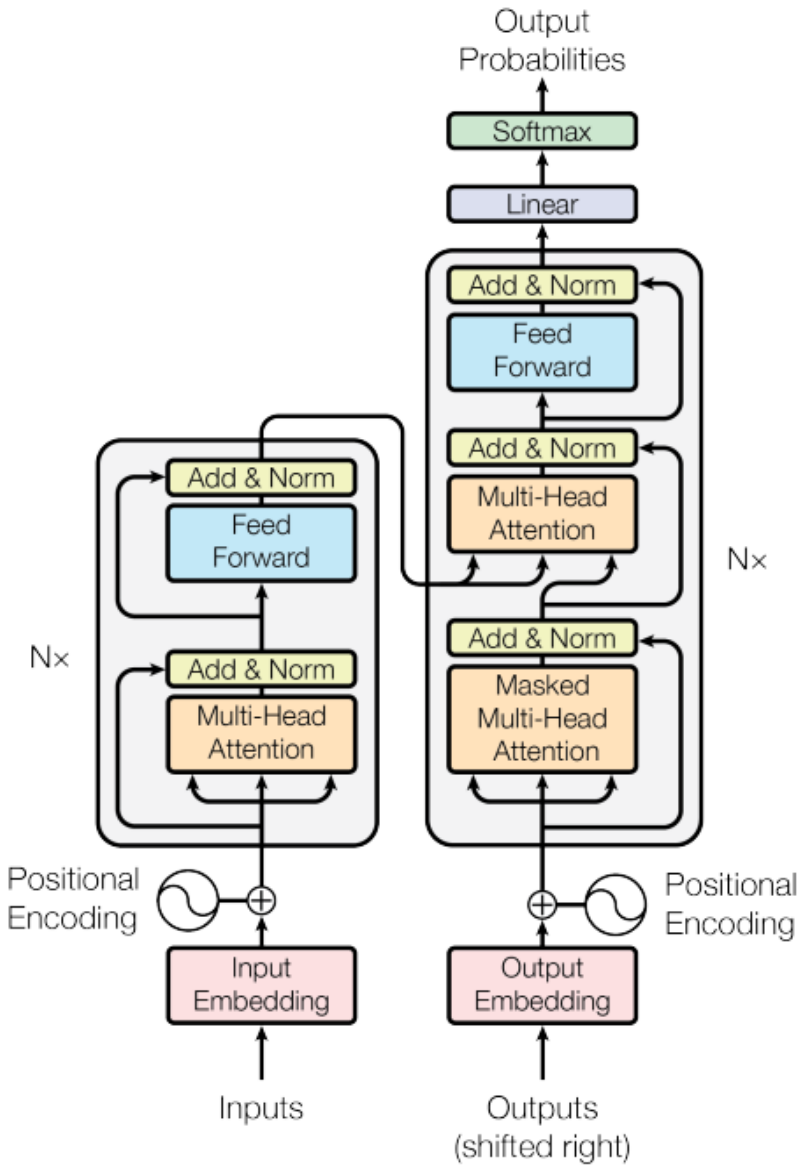
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

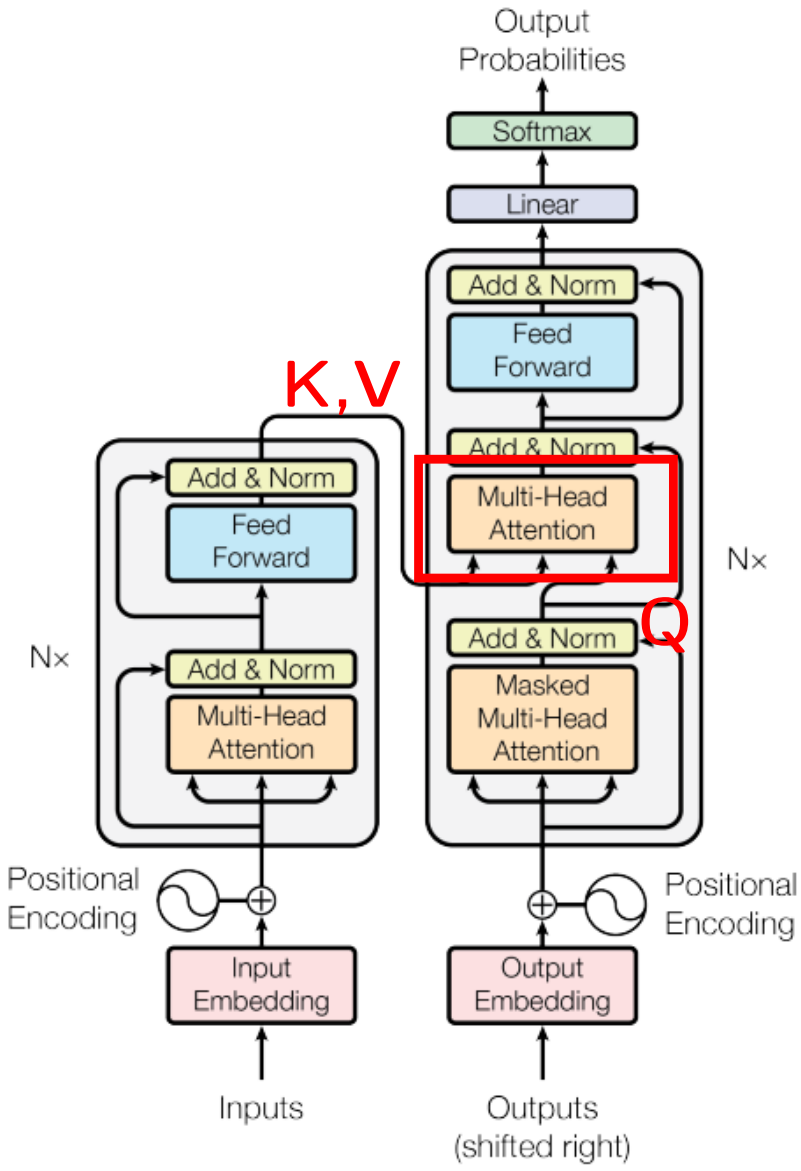
Architecture d'un *Transformer*

Remplacement des couches récurrentes ou convolutives par un mécanisme d'auto-attention (*self-attention*).
Il s'agit à nouveau d'un modèle encodeur-décodeur pour des problèmes *seq2seq*. L'encodeur prend en entrée un ensemble de *tokens*, le décodeur prend également en entrée un ensemble de *tokens* et prédit un *token* en sortie (via une distribution de probabilités sur les *tokens* possibles).



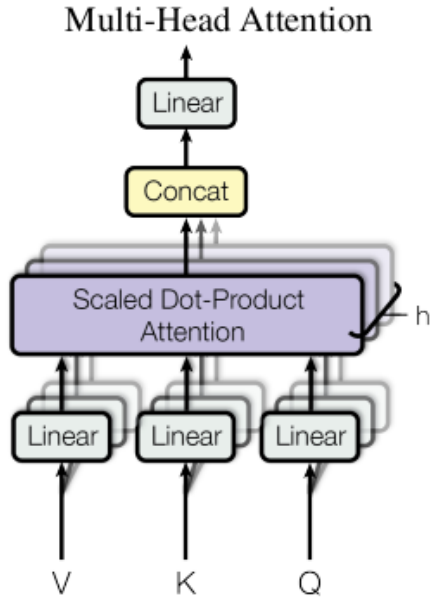
Architecture d'un *Transformer*

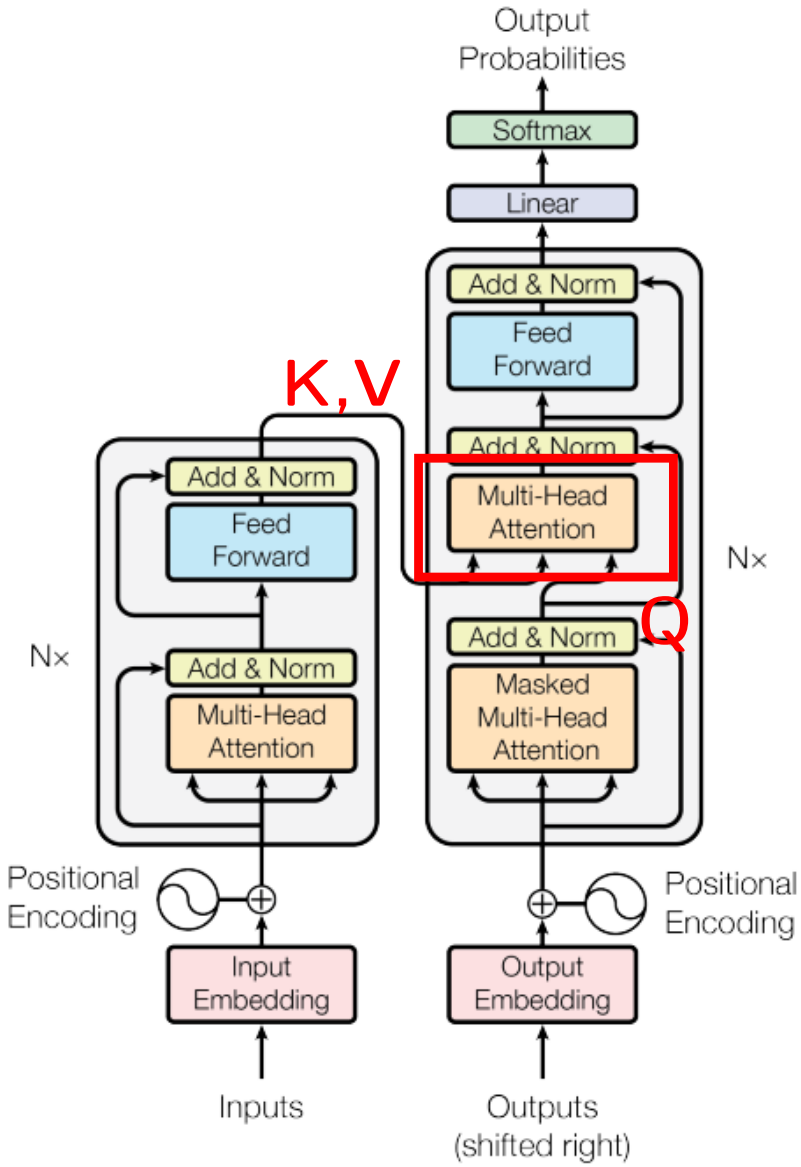
L'encodeur analyse la séquence d'entrée et produit un ensemble de clés et de valeurs.
Le décodeur analyse la séquence de sortie, produit une requête qui lui permet de se focaliser sur les parties de l'entrée les plus pertinentes à sa prédiction.



Architecture d'un *Transformer*

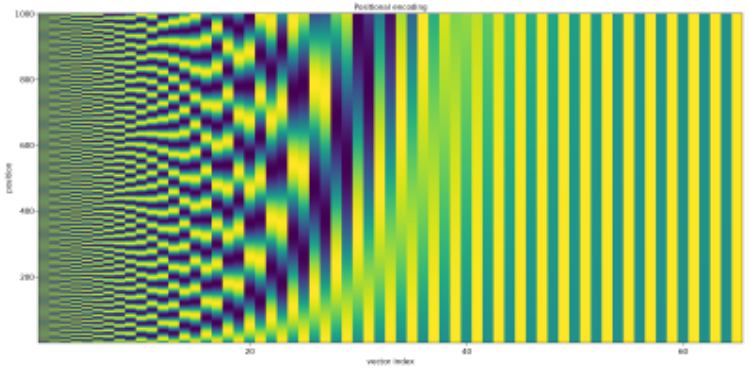
Le mécanisme d'attention est dit à "plusieurs têtes" (`multi-head`), ce qui permet de porter attention sur plusieurs éléments de la séquence avec des regards différents.

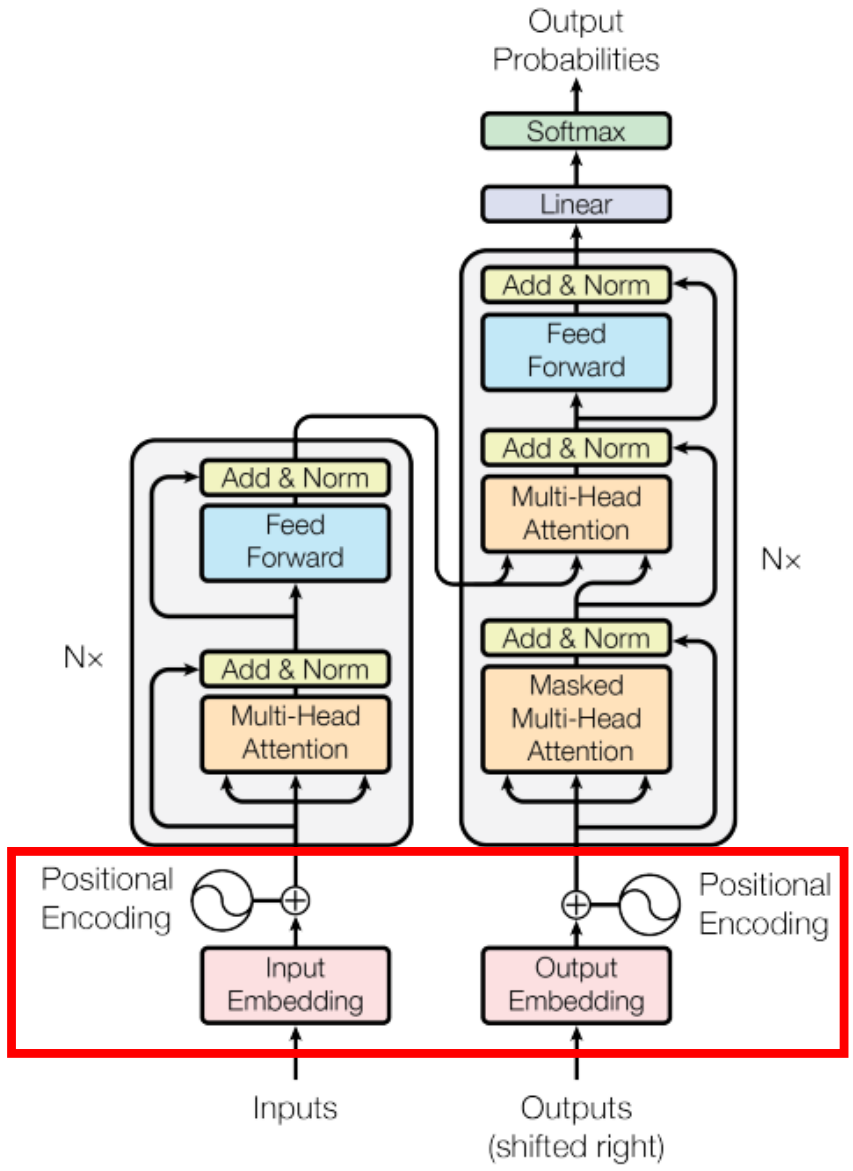




Architecture d'un *Transformer*

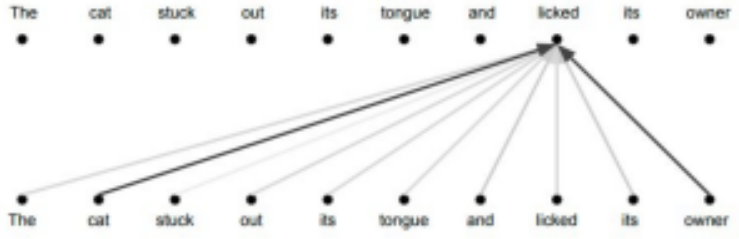
Il n'y a pas de notion d'ordre, ou de séquence, sur les tokens d'entrée. Les descripteurs associés à chaque token sont sommés à un descripteur de position (*positional encoding*), unique, qui peut également être appris (l'est souvent dans les travaux suivants).

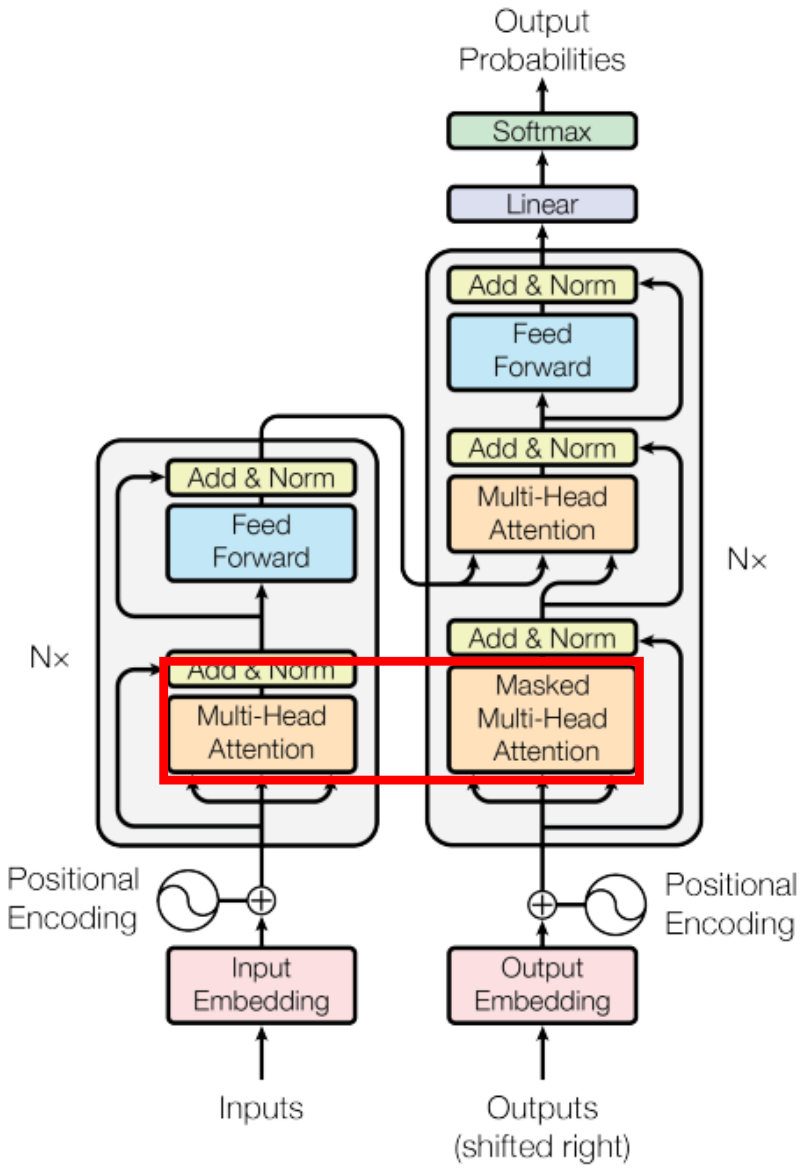




Architecture d'un *Transformer*

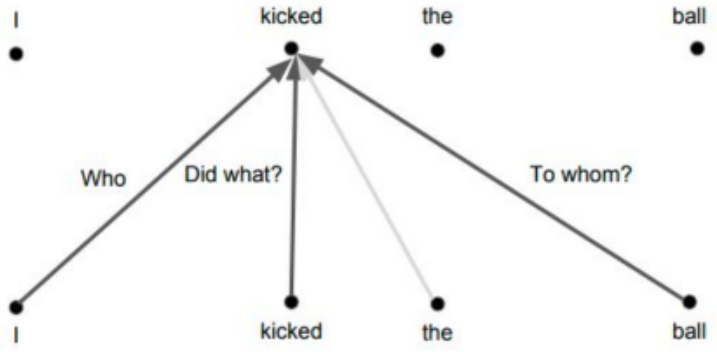
Les séquences d'entrée et de sortie sont analysées via un mécanisme d'auto-attention. Les clés, valeurs, et requêtes sont toutes générées à partir du même signal.

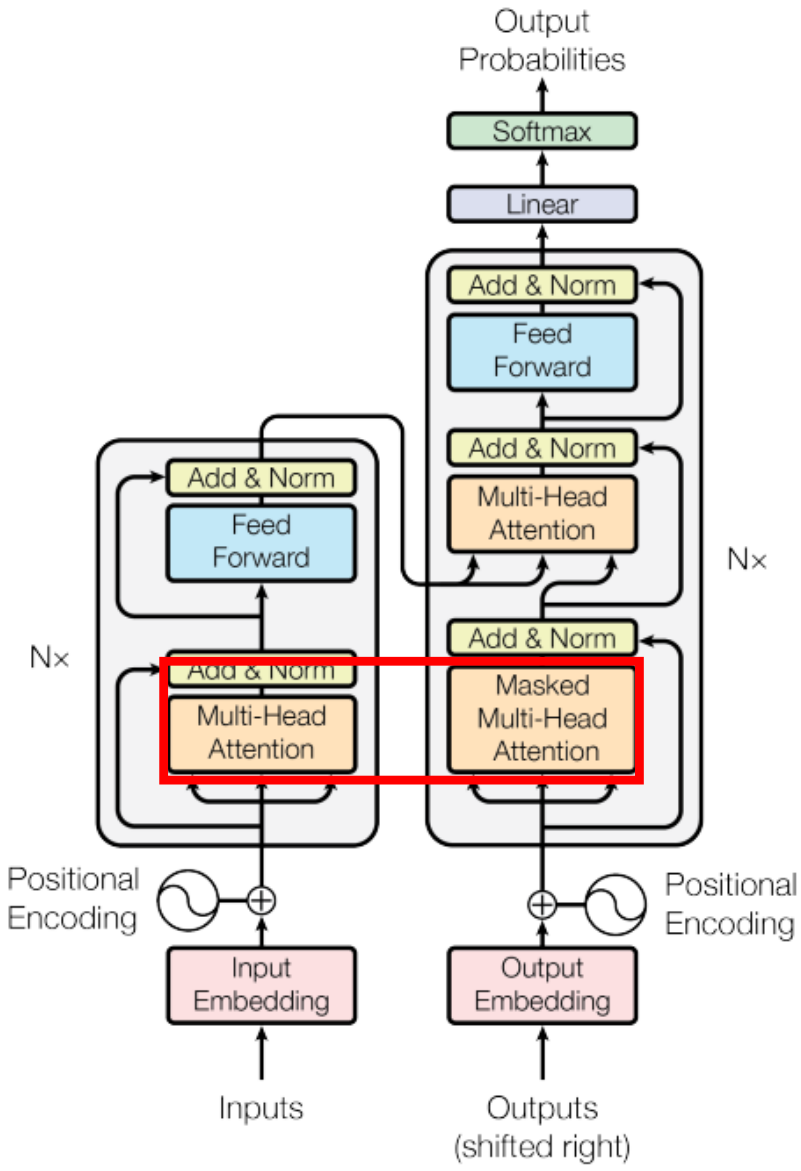




Architecture d'un *Transformer*

Les mécanismes d'attention et de *self-attention* ont "plusieurs têtes" (*Multi-head*) : plusieurs ensembles différents de clés, valeurs et requêtes permettent de porter attention à différents aspects.





Intérêt de l'auto-attention

- **Complexité plus faible** que les couches récurrentes dans les cas où la longueur de la séquence est inférieure à la dimension de la représentation maintenue ($n < d$)
- Plus de **parallélisation** possible que pour les couches récurrentes car pas de séquentialité nécessaire.
- Le **chemin minimal** dans le réseau connectant deux éléments de la séquence est beaucoup plus **court**, ce qui favorise l'apprentissage de "dépendances à long terme".

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$

où n est la longueur de la séquence, d est la dimension de la représentation, et k la dimension du noyau de convolution.

Transfert d'apprentissage

Les représentations apprises par les transformers sont transférables, et,
il est possible de pré-entraîner les transformers de manière non supervisée (cf. BERT, GPT).

Ce n'est pas le cas des réseaux récurrents, pour lesquels le transfert d'apprentissage n'a jamais réellement fonctionné.

LSTM vs. Transformers

- Les Transformers ont maintenant majoritairement remplacé les LSTM pour les tâches séquentielles (NLP, audio, vidéo).
- Les LSTM sont toujours utilisés dans deux cas de figure :
  - Séquences très longues (complexité en $O(n^2)$ des Transformers),
  - Pas de large base de données pour pré-entraîner les Transformers (sur de petits échantillons et sans pré-entraînement, LSTM $>$ Transformers)

Plan du cours

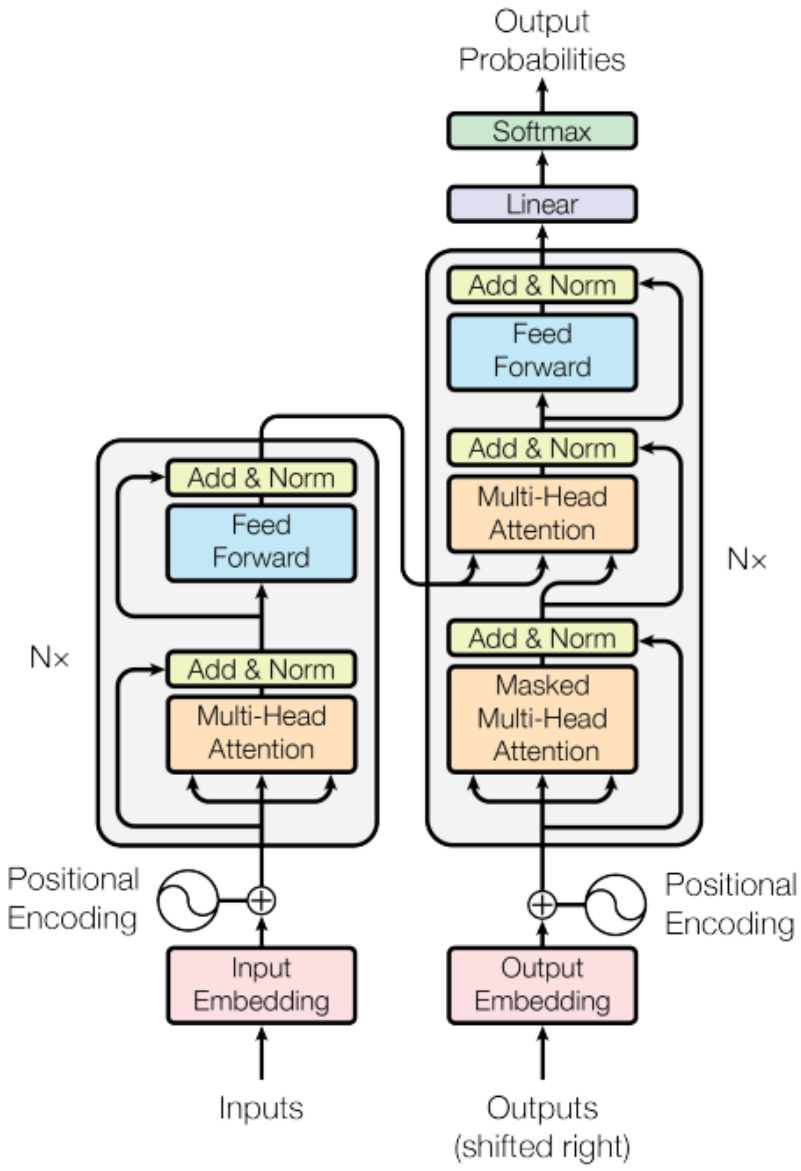
1. Recherche d'information
2. Mécanisme d'attention
3. Transformers
4. Applications et évolutions récentes

Architecture d'un *Transformer*

Dans l'architecture ci-contre :

- Nombre de répétitions de chaque bloc : $N = 6$
- 8 têtes d'attention en parallèle
- Dimension maintenue pour la représentation interne des *tokens* : $d = 512$

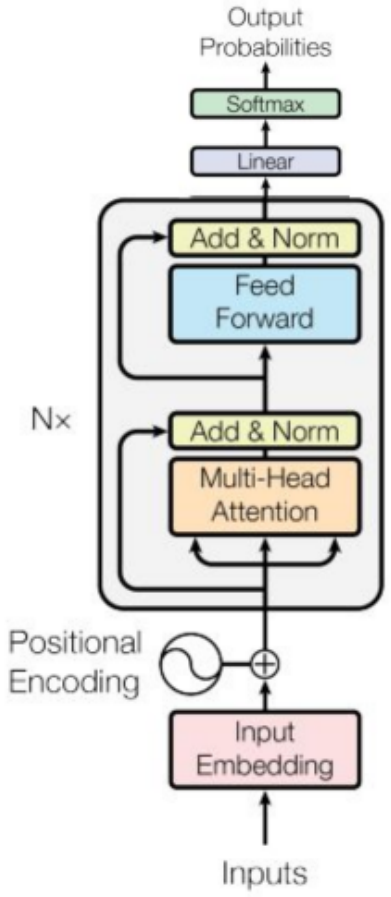
65 millions de paramètres



Google BERT

- Architecture ressemblant à l'encodeur du Transformer originel : l'attention peut être portée à tous les *tokens* de la séquence (bidirectionnel).
- Pré-entraînement non-supervisé sur de larges corpus de textes, avec masquage de mots dans la séquence.

Version française : CAMEMBERT (Meta)



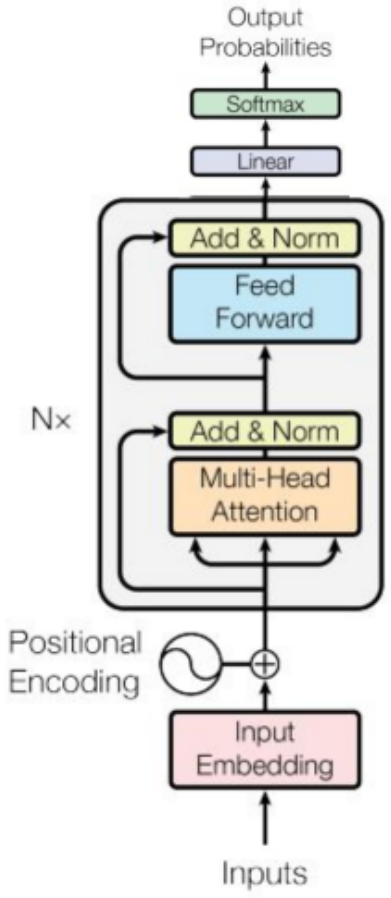
[Devlin 2018] BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding

[Martin 2019] CamemBERT : a tasty French language model

Google BERT

- Nombre de répétitions de chaque bloc : $N = 24$
- 16 têtes d'attention en parallèle
- Dimension maintenue pour la représentation interne des *tokens* : $d = 1024$

340 millions de paramètres

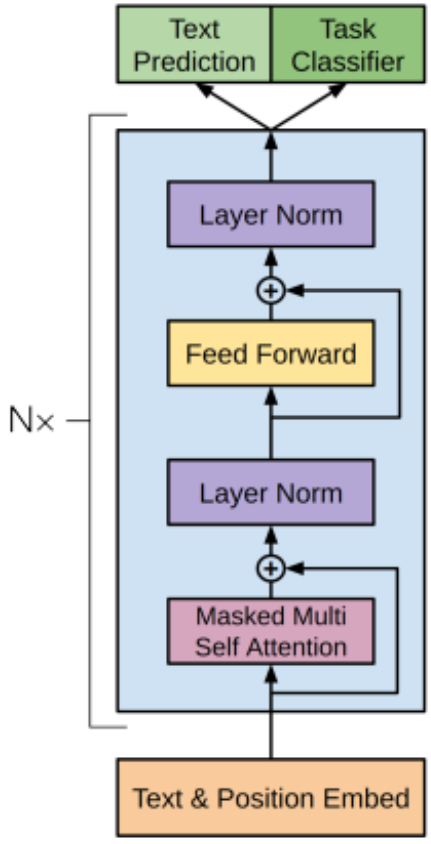


[Devlin 2018] BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding

OpenAI GPT

- Architecture ressemblant au décodeur du Transformer originel : l'attention ne peut être portée qu'aux tokens précédents dans la séquence.
- Pré-entraînement non-supervisé sur un large corpus de textes (BooksCorpus : 11000 livres non publiés, 1 milliard de mots).

A obtenu les meilleures performances de l'état de l'art sur 9 tâches simultanément (Similarité entre phrases, classification, réponse à une question, etc.)

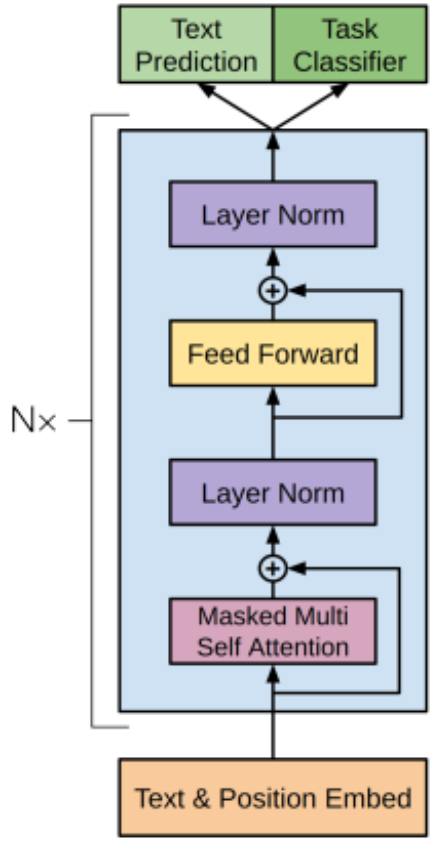


[Radford 2018] Improving Language Understanding by Generative Pre-Training

OpenAI GPT

- Nombre de répétitions de chaque bloc : $N = 12$
- 12 têtes d'attention en parallèle
- Dimension maintenue pour la représentation interne des *tokens* : $d = 768$

117 millions de paramètres

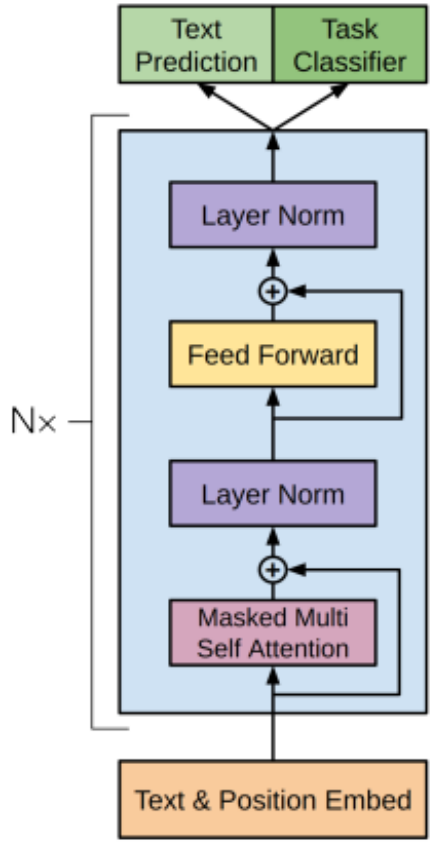


[Radford 2018] Improving Language Understanding by Generative Pre-Training

OpenAI GPT2

- Nombre de répétitions de chaque bloc : $N = 48$
- 12 têtes d'attention en parallèle
- Dimension maintenue pour la représentation interne des *tokens* : $d = 1600$

1,5 milliards de paramètres



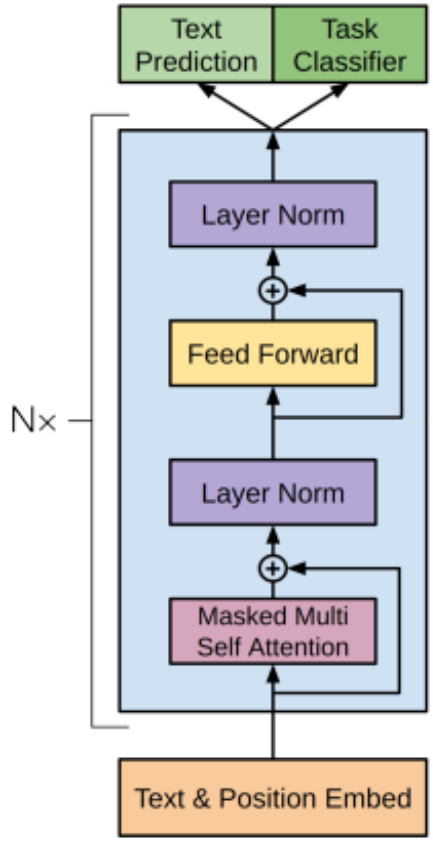
Pré-entraînement sur WebText, 8 millions de pages Web représentant plus de 40 Go de texte

[Radford 2019] Language Models are unsupervised multitask learners

OpenAI GPT3

- Nombre de répétitions de chaque bloc : $N = 96$
- 96 têtes d'attention en parallèle
- Dimension maintenue pour la représentation interne des *tokens* : $d = 12888$ (! !)

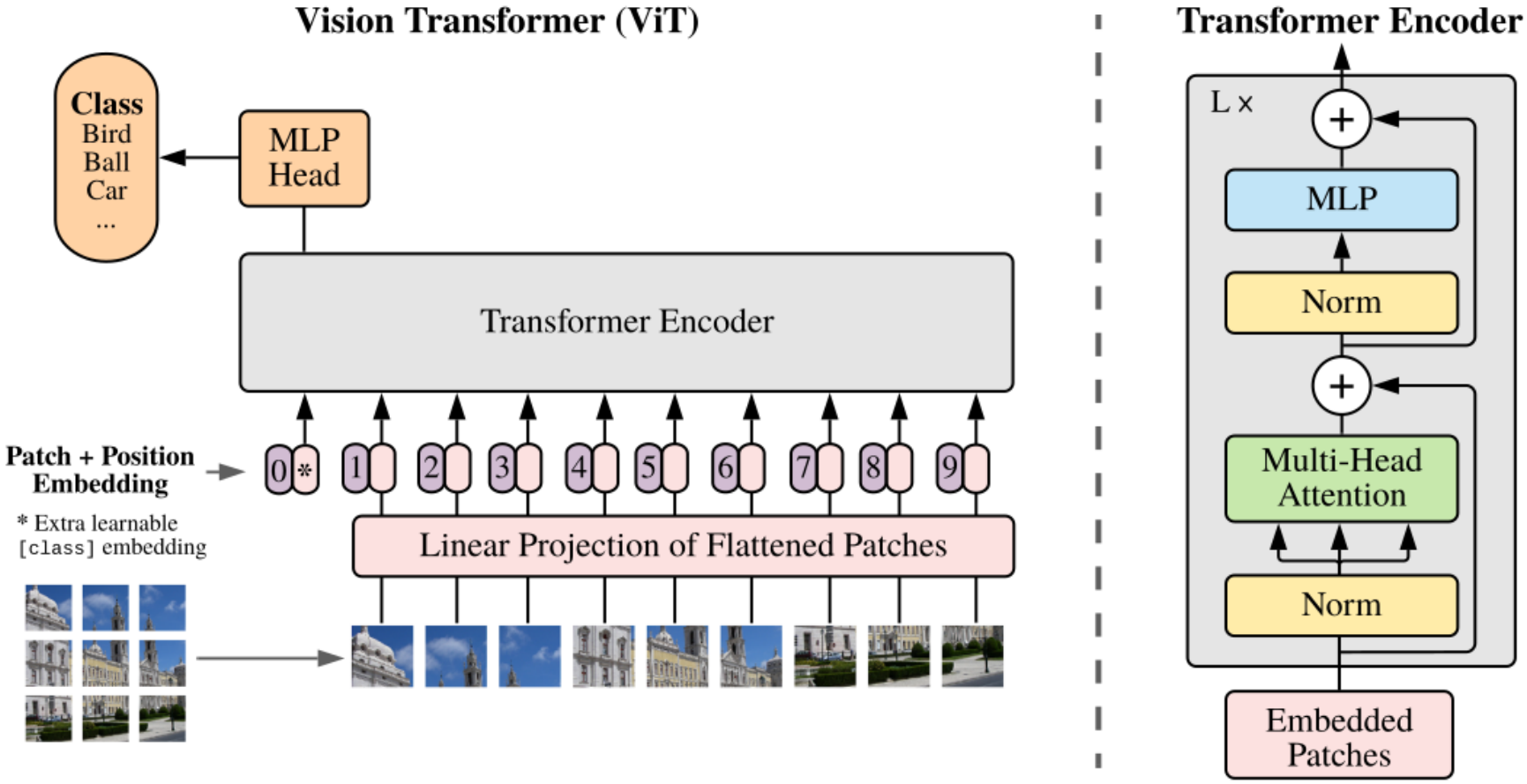
175 milliards de paramètres



Pré-entraînement sur CommonCrawl (45 To de texte), WebText, des bases de données de livres, ainsi que Wikipedia (qui ne représente que 3% du total des données)

[Brown 2020] Language models are few shot learners

Vision Transformers (ViT)



[Dosovitskiy 2020] An Image is Worth 16×16 Words : Transformers for Image Recognition at Scale