# Data Challenge

NLP – Original Transformer

Ecole polytechnique
Chaire BAFB sponsorisée par Natixis

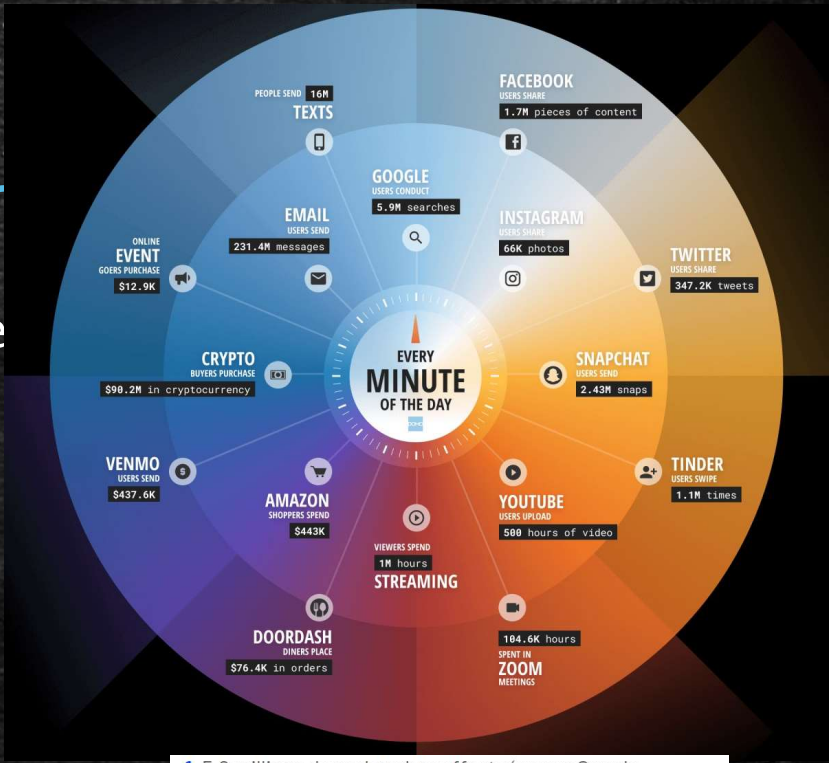
# NLP

1. Intérêts
2. Pipelines
3. Original Transformer
4. Extensions, Mise en route et ressources

# 1. Intérêts

- Plus de 250 M données en une minute
- Pas possible d'analyser manuellement une faible partie des données
- Essentiellement des données au format texte

➔ Conception d'outils et de méthodes (pipelines) pour tirer l'information des données des réseaux sociaux




1. 5,9 millions de recherches effectuées sur Google,
2. 1,7 million de contenus partagés sur Facebook,
3. 66 000 photos partagées sur Instagram,
4. 347 000 tweets postés sur Twitter,
5. 2,4 millions de snaps envoyés sur Snapchat,
6. 500 heures de vidéo téléchargées sur YouTube,
7. 443 000 dollars dépensés sur Amazon,
8. 16 millions de messages textes envoyés,
9. 231 millions d'emails envoyés,
10. 90 millions de dollars dépensés dans les crypto-monnaies.

https://www.blogdumoderateur.com/une-minute-internet-2022/

# 1. Intérêts

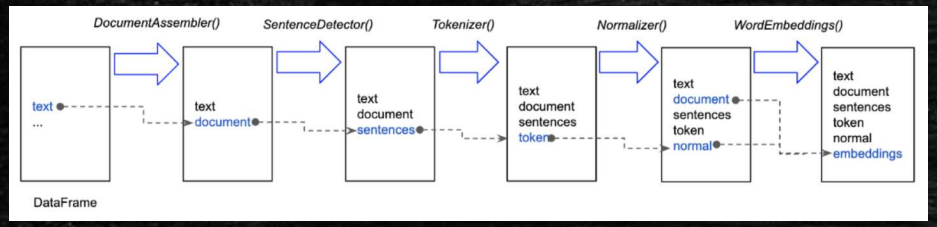
- Utilisation traitement classique pour analyse des réseaux sociaux
  - Détection de sujets tendance
    ➔ stratégie médias, politiques….
  - Opinions
    ➔ Informations sur produit, service, politique
  - Sentiment Analysis
    ➔Information marketing pour les marques à propos de  leurs services, leurs produits…
  - Rumeur / fake news
    ➔ Utilisation à mauvais escient pour diffuser des fausses informations par groupes, stratèges… afin d'influencer sur les populations
  - Filtrage de contenus
    ➔ Identification et filtrage des contenus inappropriés (nudité, blasphème, racisme, les menaces, etc.
  - Service client
    ➔ Identification des problèmes, des plaintes, classification  des problèmes

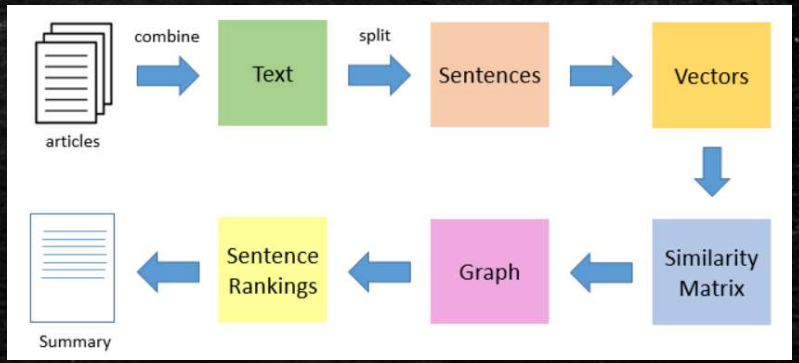
# 2. Pipelines

- Définition :

  - Utilisation successive de plusieurs étapes ordonnées ou modèles de traitement de texte pour effectuer une tâche spécifique. Chaque modèle dans le pipeline prend en entrée le texte et renvoie une sortie qui est utilisée par le modèle suivant.

  - Par exemple,
    - pipeline de traitement de texte pour la traduction automatique
    - pipeline  de réponse automatique aux questions
    - pipeline de résumé automatique de texte.



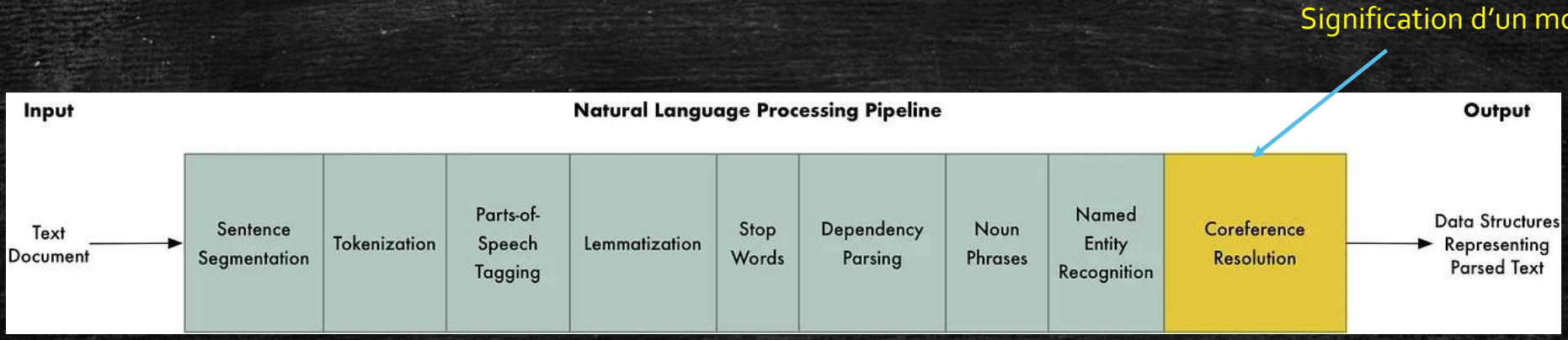




https://towardsdatascience.com/text-classification-in-spark-nlp-with-bert-and-universal-sentence-encoders-e644d618ca32

https://www.kaggle.com/getting-started/153771

# 2. Pipelines

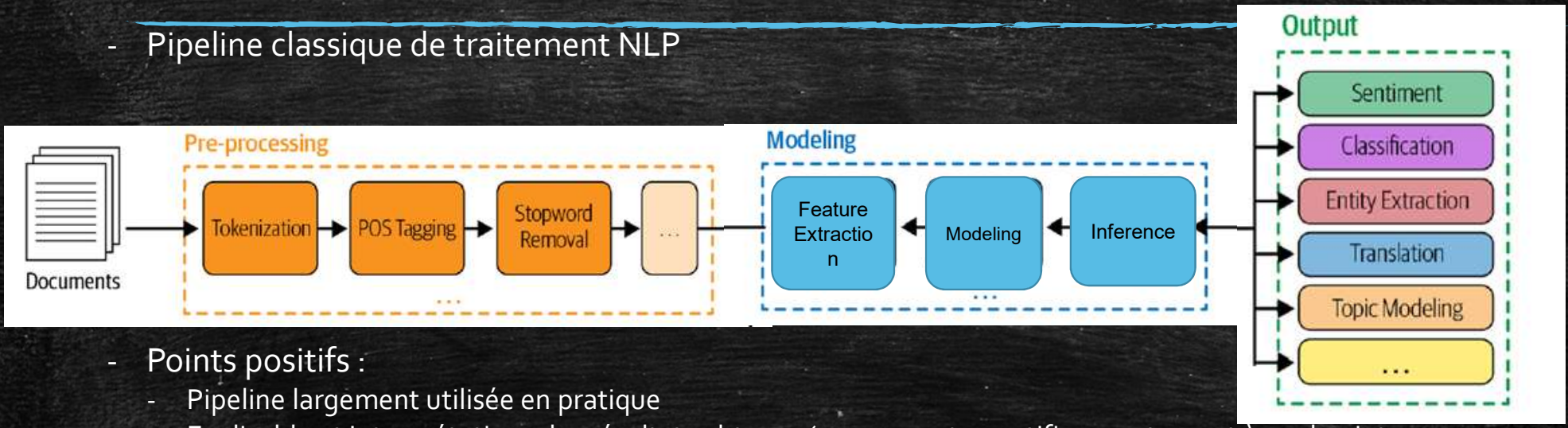
- Définition :
  - Pipeline personnalisée / objectif, tâche




https://medium.com/@ageitgey/natural-language-processing-is-fun-9a0bff37854e

# 2. Pipelines

- Pipeline classique de traitement NLP




- Points positifs :
  - Pipeline largement utilisée en pratique
  - Explicable et Interprétations des résultats obtenus (ex : on peut quantifier exactement à quel point chaque caractéristique influence la prédiction du modèle)
  - Assez rapide
- Points négatifs :
  - Beaucoup de constructions d'étapes manuelles
  - Cycle de vie de développement de la chaine
  - Influence sur performance du modèle

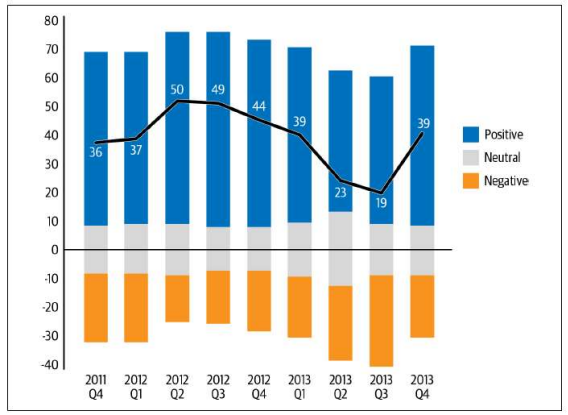
https://www.oreilly.com/library/view/practical-natural-language/9781492054047/

# 2. Pipelines

- Résultats de l'utilisation traitement classique de pipeline

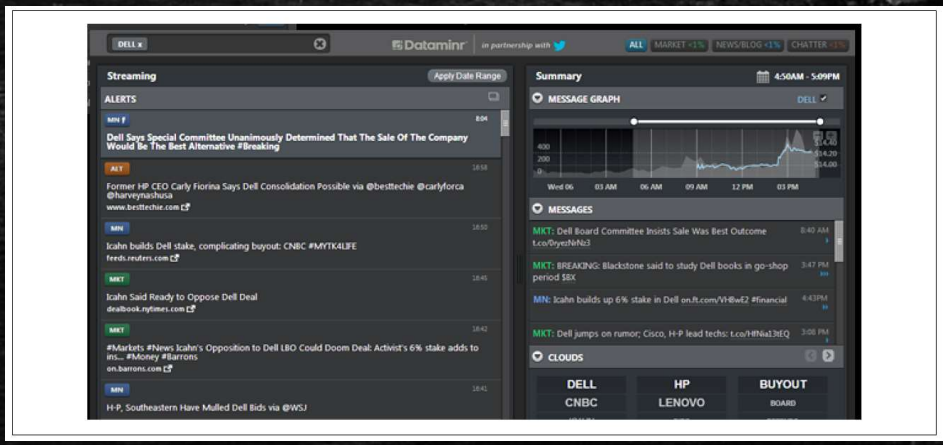
Avis sur produit

Evolution sentiment





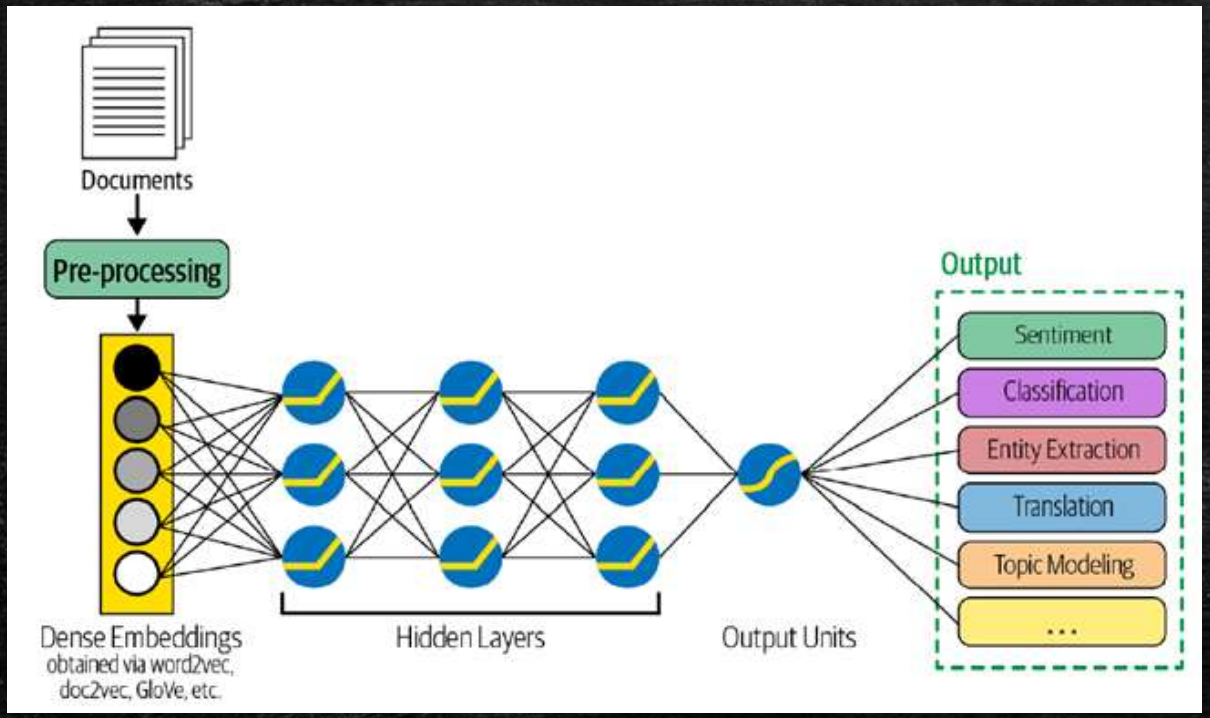
Information pour plateforme de trading




https://www.oreilly.com/library/view/practical-natural-language/9781492054047/

# 2. Pipelines

- Pipeline Deep Learning NLP




RNN, CNN, LSTM, Transformer,…

https://www.oreilly.com/library/view/practical-natural-language/9781492054047/

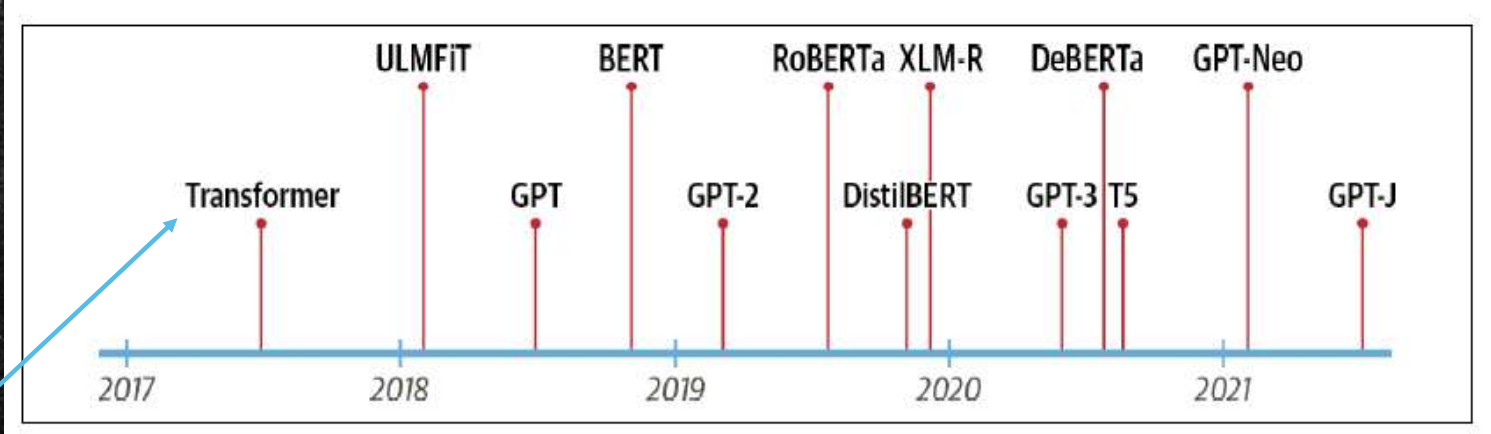
-Points positifs :
- Peu d'interventions manuelles (pré-processing)
- Modèle apprend à partir des données

-Points négatifs :
- Perte d'interprétabilité
- Difficulté à expliquer la prédiction (ex : pourquoi ce message est considéré comme spam)
- Difficile à utiliser opérationnellement dans un contexte industriel, commercial…
- gourmand en ressources

-Choix du type de pipeline dépend de la problématique, de l'objectif et des données
-Ex : DL NLP adapté pour traduction automatique, Classique NLP pour analyse marketing

# 2. Pipelines

- Historique des transformers




Original Transformer

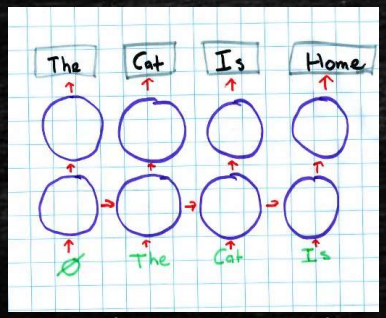
Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

# 2. Pipelines

- Article de référence : Original transformer
  - **Attention Is All You Need**, Vaswani and Al., 2017, Google

  - *The dominant sequence transduction models are based on complex recurrent or convolutional neural networks in an encoder-decoder configuration. 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.*

Récurrence dans les RNN




**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

**Abstract**

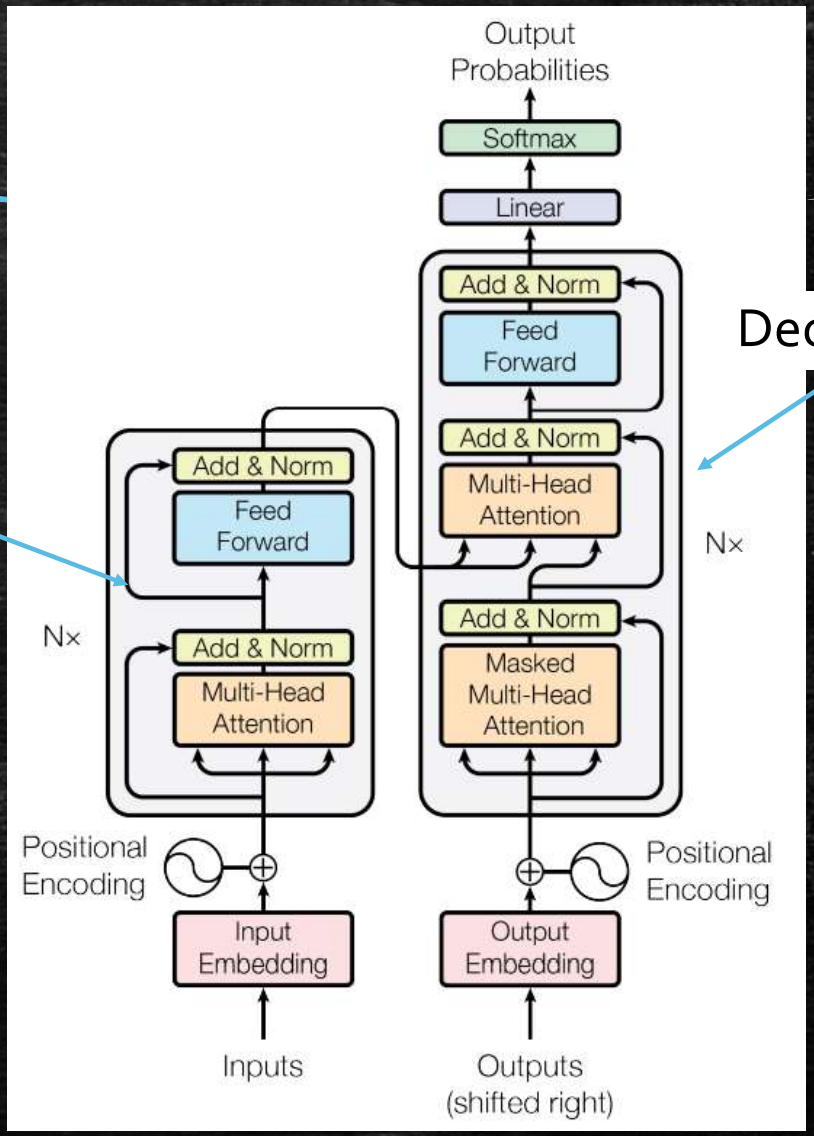
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, 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.

https://arxiv.org/abs/1706.03762

https://medium.com/machine-learning-at-petiteprogrammer/sampling-strategies-for-recurrent-neural-networks-9aea02a6616f

# 3. Original Transformer

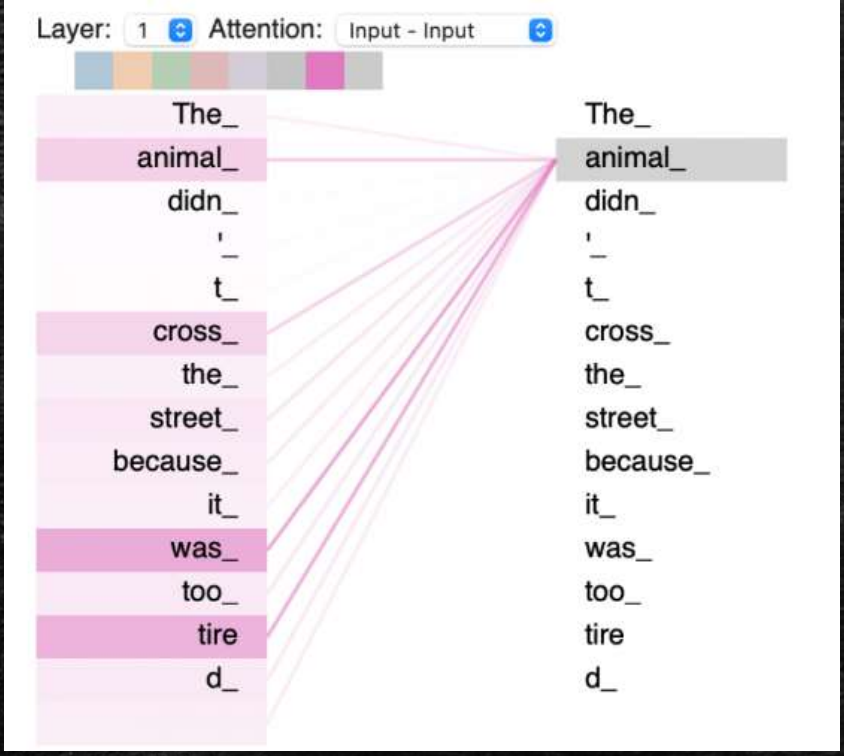
- Architecture proposée (2017)
- Pas de réseaux RNN, CNN…
- Partie Encoder
  - 1 couche Attention et 1 couche Feed Forward
- Partie Decoder
  - 2 couches Attention et 1 couche FF
- Mécanisme d'attention
  - Remplace la récurrence
  - Identifie la liaison d'un mot aux autres mots




Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

# 3. Original Transformer

- Mécanisme d'attention (illustration)




https://inside-machinelearning.com/mecanisme-attention/

**Mécanisme d'attention :**

- Visualisation des liens qui unissent les mots de la phrase

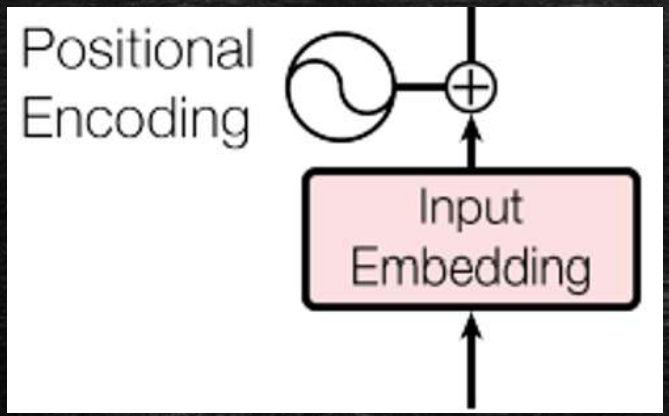
-« animal » avec « crossed », « was », « tired »

- Liaison même avec ds mots éloignés dans la phrase ➔ compréhension de la relation

- Détection de contexte ici l'animal n'a pas travasersé la rue car il était fatigué
➔compréhension globale de la phrase

# 3. Original Transformer

- Encoder : contexte basé sur la position et l'ordre des mots
- Hypothèse de conception du Transformer : les sorties de chaque sous-couche sont de même dimension $d_{model} = 512$.




Chaque mot est représenté dans l'espace d512 (word2vec…)

[The, cat, did,n,',t, cross, the, street, because, it, was, too, tired]

Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

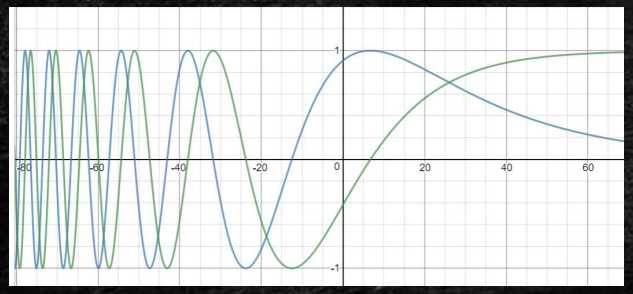
# 3. Original Transformer

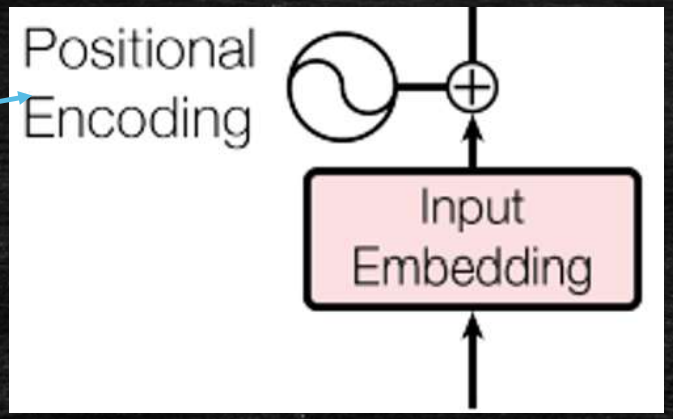
- Encoder

- Hypothèse de conception du Transformer : les sorties de chaque sous-couche sont de même dimension $d_{model} = 512$.

$$PE_{(pos\ 2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

$$PE_{(pos\ 2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$






Chaque mot de la phrase est encodé selon sa position Ex : cat 2, street 9)

Position paire (i) : calcul en sinus(i)

Position impaire (2i +1): calcul en cosinus

Chaque vecteur de position est de la même dimension que celle de l'Embedding

➔Importance du PE (prise en compte de l'ordre des mots)

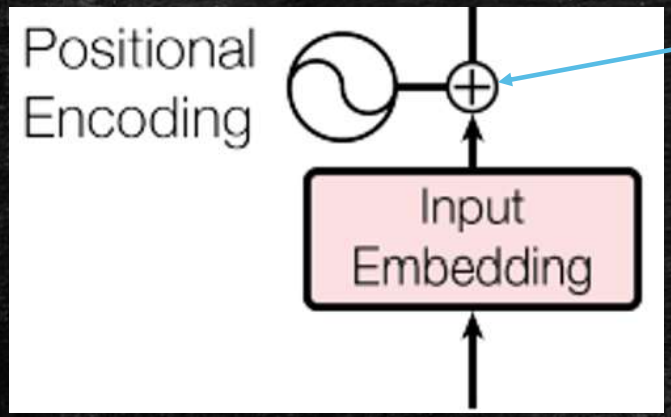
[The, cat, did,n,`,t, cross, the, street, because, it, was, too, tired]

Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

# 3. Original Transformer

- Encoder

- Hypothèse de conception du Transformer : les sorties de chaque sous-couche sont de même dimension $d_{model} = 512$.




Positional Vector = Input Embedding + Positional Encoding

➔ Encodage de la place de chaque élément dans la pharase (séquence)

➔ Comme la longueur des phrases n'est pas prédéterminée, les fonctions sinusoïdales et cosinusoïdales (valeurs comprises entre e o et 1) permettent de modifier les dimensions embeddings de chaque mot.

➔ Calcul efectué de même dimension (d = 512)

A la sortie de cette étape, matrice (nb mots x 512)

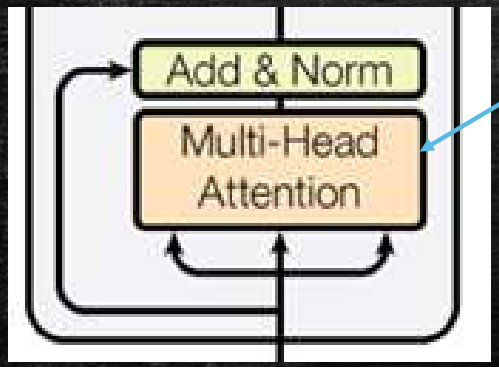
[The, cat, did,n,`,t, cross, the, street, because, it, was, too, tired]

Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

# 3. Original Transformer

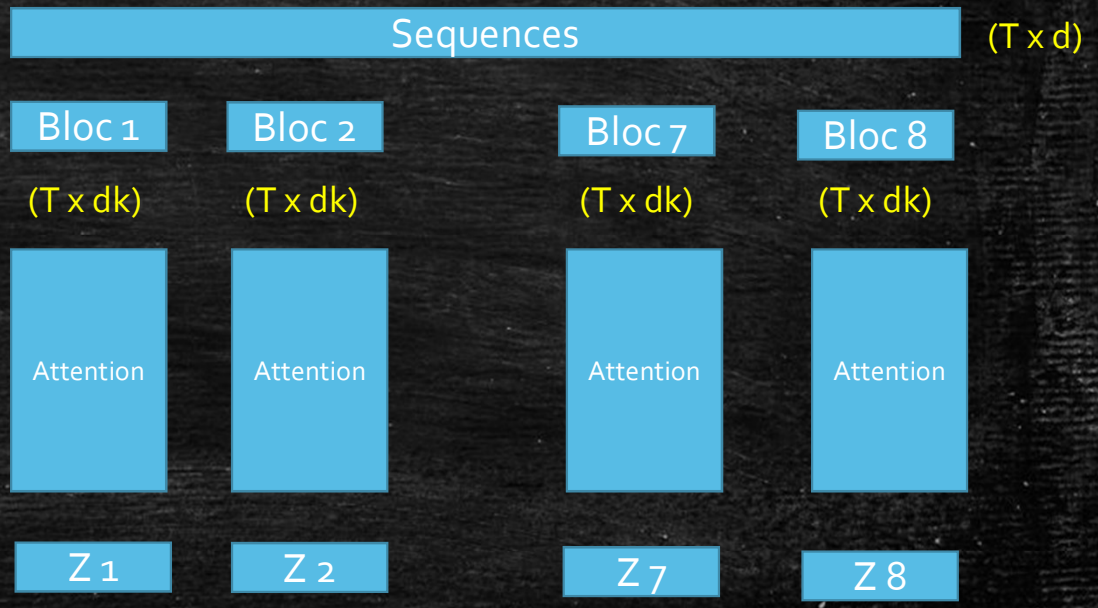
- Architecture proposée (2017)

Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.




matrice (T x d)
(d = 512)

Les séquences (T x 512) sont décomposées en k (8) blocs de dimension dk (64)




Objectif :
Faire plusieurs mécanismes d'attention sur plusieurs dimensions (heads)
Est-ce que le mot « cat » est plus lié au mot « it » ? On répète plusieurs fois le mécanisme sur plusieurs « heads »

[The, cat, did,n,',t, cross, the, street, because, it, was, too, tired]

# 3. Original Transformer

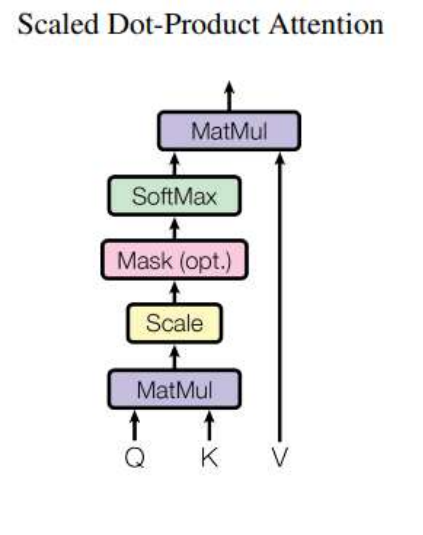
- Mécanisme d'attention
  - Utilisé à plusieurs couches dans le Transformer pour permettre au modèle de prendre en compte les relations entre les différentes parties de l'entrée, à se concentrer sur les parties importantes de l'entrée (séquence)

  - Pendant l'entraînement, le modèle apprend à pondérer les différentes parties de l'entrée pour calculer une représentation contextuelle pour chaque élément de l'entrée.

Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

# 3. Original Transformer

- Principe de self attention
  - Idée : récupération de l'information de chaque token sur lui-même et avec les autres tokens de la séquence (≠ récupération de manière séquentielle
  - Récupération « self attention »
    - Attention sur chaque token avec lui même et les autres tokens dans la séquence




A l'intérieur de chaque « head », représentations de chaque token en 3 représentations :

- Vecteur Q (Query) de dimension de dk (64)
- Vecteur K (Key) de dimension de dk
- Vecteur V (Value) de dimension dk

[The, cat, did,n,',t, cross, the, street, because, it, was, too, tired]

NB : il faut initializer des matrices de poids Qw, Kw, Vw pour obtenir les matrices Q, K et V

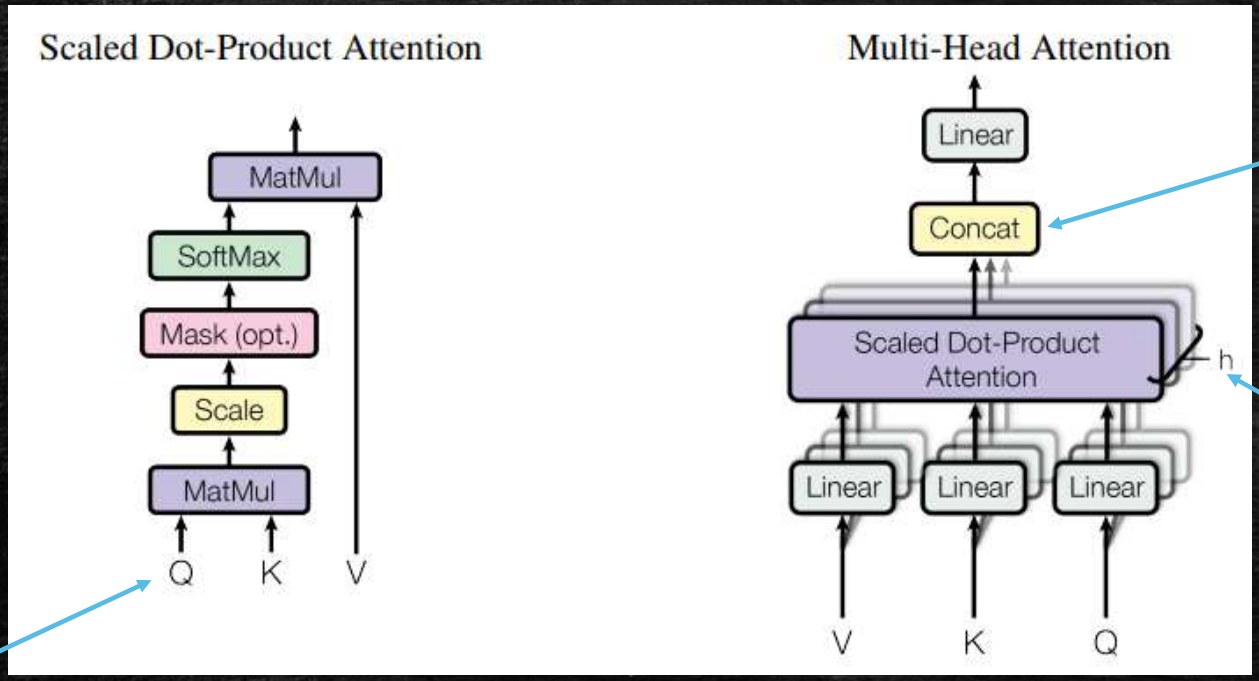
**Attention = "Scaled Dot-Product Attention"**

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

Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

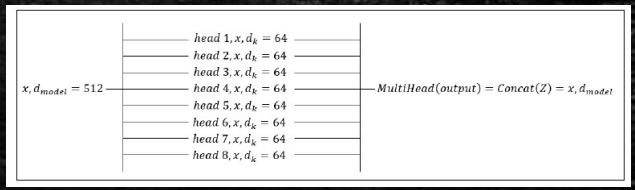
# 3. Original Transformer

- Multiple-Head Attention
  - Passage du Self Attention au Multi-Head Attention




Concaténation des Self-Attention

Self-Attention appliqué sur chaque head

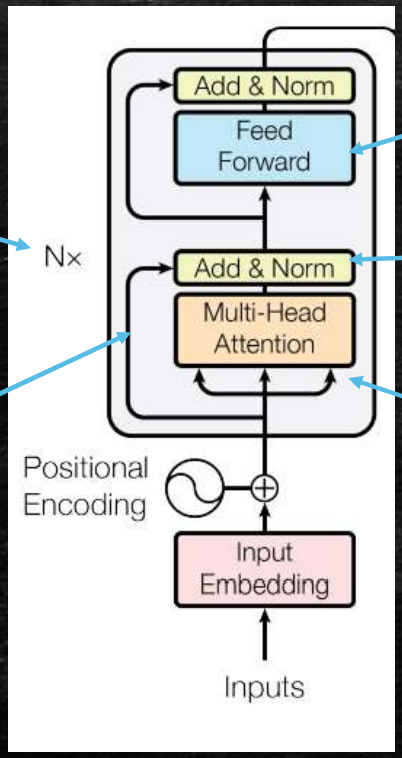



Pour une « head »

Chaque « head » de l'attention a ses propres matrices de poids Qw, Kw, Vw

# 3. Original Transformer

- Encoder
  - Descriptif des opérations de passage entre blocs et sous-couches




Le processus est répété N fois

Couches de neurones formels avec une ReLU comme fonction d'activation

Addidion vectorielle des outputs (z :(T, dmodel)) (et normalisation vectorielle

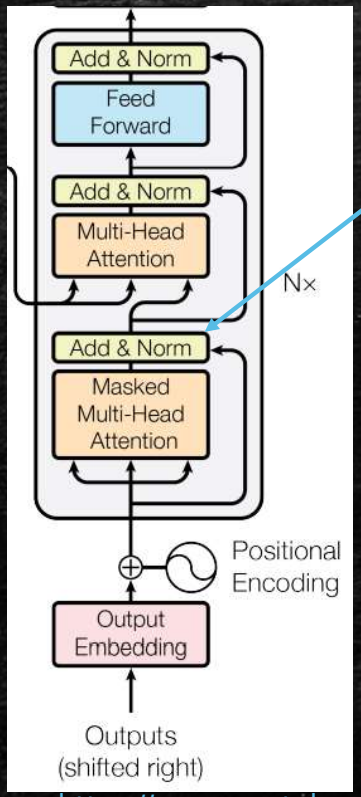
$$\text{LayerNorm}(v) = \gamma \frac{v - \mu}{\sigma} + \beta$$

Ajout d'une liaison sans passer par la phase Attention si cette dernière n'a pas bien fonctionnée

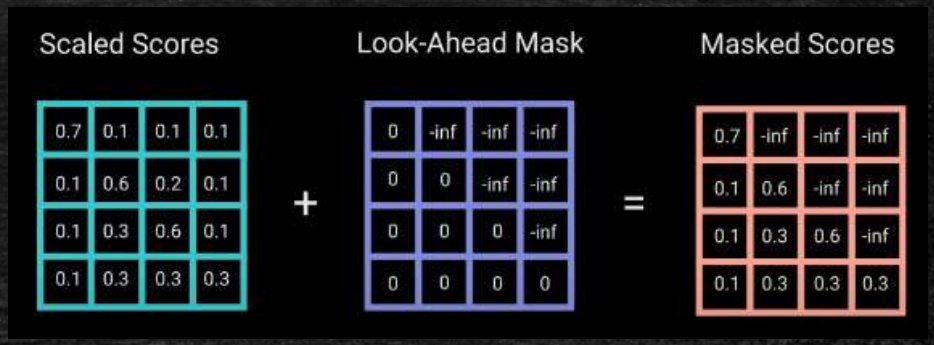
Flèches représentant les matrices Q, K et V (Ex : Q = Inputs * Qw)
**Qw : (dmodel, dk)**

# 3. Original Transformer

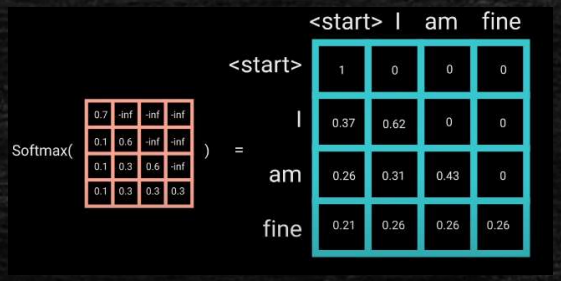
- Decoder
  - Descriptif des opérations de passage entre blocs et sous-couches




Le but est d'empêcher au Decoder de ne pas prendre en compte les tokens futurs
Pour ne pas accéder aux tokens futurs, utilisation Masking Method (on ne veut pas calculer des attentions sur les tokens futurs). Accès uniquement aux tokens précédents

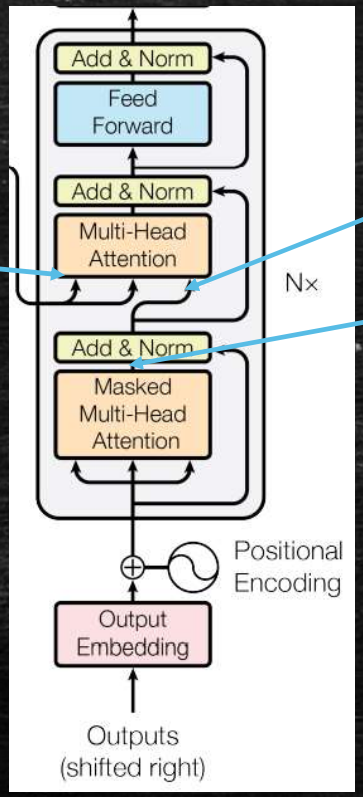

Ensuite application d'une fonction softMax sur les Masked Scores




https://www.youtube.com/watch?v=4Bdc55j80l8

# 3. Original Transformer

- Decoder
  - Descriptif des opérations de passage entre blocs et sous-couches



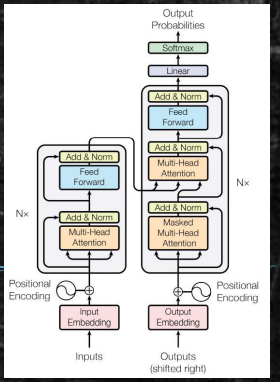

Dans la 2ème phase d'Attention, incorporation des inputs de l'Encoder (matrices K et V)

Sélection des matrices K et V qui permettent d'améliore la prédiction des motes à traduire

Matrice Q du Decoder

En sortie de la 1ère phase d'Attention, Masked vector
Indication au Decoder sur quel token porter l'Attention

# 3. Original Transformer



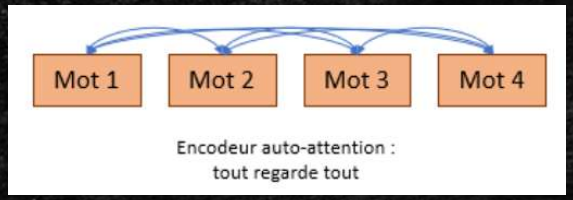

- Decoder
  - Descriptif des opérations de passage entre blocs et sous-couches

Encoder - Decoder


Prédiction du mot masqué.
Le Decoder regarde les attentions des mots de l'Encoder.
Le Decoder doit apprendre que le mot caché est le Mot 3 (utilisation de la fonction Attention)
Modification de la matrice Q pour le mot à prédire ait une grande valeur de softmax

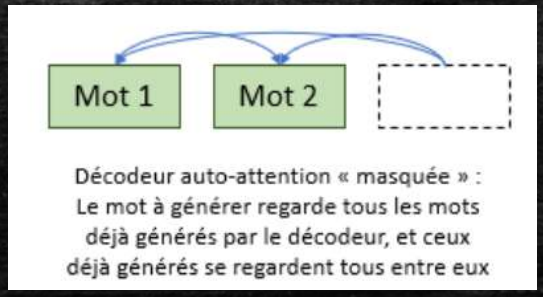
$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Partie Encoder




On regarde tous les attentions entre les tokens embedded

Decoder




On regarde les attentions entre les tokens embedded non masqués

# 3. Original Transformer

- Encoder-Decoder
  - Long apprentissage
  - Transformer original entrainé sur un ensemble de données anglais-allemand de 4,5 millions de phrases et un ensemble de données anglais-français de 36 millions de phrases.
  - Entrainement du modèle de base : 12 heures avec une machine avec 8 GPU NVIDIA P100. Les grands modèles ont nécessité 3,5 jours.
  -
  - Transformateur original a surpassé tous les modèles de traduction automatique précédents avec un score BLEU de 41,8 (Score BLEU Bilingue Evaluation Understudy).
  - Autres stratégies d'optimisation pour améliorer les performances du transformateur....

# 4. Extensions, Mise en route et ressources

- Implémentation
- Environnement “Hugging Face” : utilisation du modèle entrainé

- 3 lignes de code :
  - !pip -qq install transformers

1. from transformers import pipeline
2. translator = pipeline("translation_en_to_fr")
3. print(translator("It is easy to translate languages with transformers", max_length=40))

```
[{'translation_text': 'Il est facile de traduire des langues avec des
transformateurs.'}]
```

# 4. Extensions, Mise en route et ressources

- Nouveaux transromers
  - Bert  Base
    - Encoder stack : 12
    - Dmodel = 768
    - Heads : 12 de dimension dk = 64
  - Bert Large
    - Encoder stack : 24
    - Dmodel : 1024
    - Heads : 16 de dimension dk = 64

# 4. Extensions, Mise en route et ressources

- Ressources
  - "Attention Is All You Need" - Publication de Google en 2017 a introduit le modèle de transformer, qui est devenu un pilier de nombreux modèles de NLP modernes.
  - "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" - Publication de Google en 2018 décrit le modèle BERT (Bidirectional Encoder Representations from Transformers), qui a été l'un des modèles les plus performants pour de nombreuses tâches de NLP.
  - "RoBERTa: A Robustly Optimized BERT Pretraining Approach" - Publication de Facebook en 2019 décrit une version optimisée du modèle BERT, appelée RoBERTa, qui a surpassé les performances de BERT sur de nombreuses tâches de NLP.
  - "Transformers: State-of-the-art Natural Language Processing" - Livre de 2020, écrit par un groupe d'auteurs, fournit une introduction complète aux modèles de transformers pour le NLP, y compris les concepts de base, les architectures de modèles, les méthodes d'entraînement et les applications.
  - "Deep Learning for Natural Language Processing" - Livre de 2018, écrit par Palash Goyal, Sumit Pandey et Karan Jain, fournit une introduction complète à la NLP et aux techniques de deep learning pour le traitement de texte, y compris une section dédiée aux modèles de transformers.