L'attention est tout ce dont vous avez besoin

Ashish Vaswani[1] Google Cerveau avaswani@google.com
Noam Shazeer[1] Google Cerveau noam@google.com
Niki Parmar[1] Recherche Google nikip@google.com
Jakob Uszkoreit[1] Recherche Google usz@google.com

Llion Jones[1] Recherche Google Illion@google.com
Aidan N. Gomez[*] [†] Université de Toronto Aiden@cs.toronto.edu
Łukasz Kaiser[1] Google Cerveau lukaszkaiser@google.com

Illia Polosukhin[*] [*] Illia.polosukhin@gmail.com

Résumé

Les modèles dominants de transduction de séquence sont basés sur des
réseaux neuronaux convolutionnels comprenant un codeur et un décodeur.
des modèles performants relient également l'encodeur et le décodeur à travers une attention
Nous proposons une nouvelle architecture de réseau simple, le Transformer,
basée uniquement sur les mécanismes d'attention, la distribution avec récurrence et les convolutions
entièrement. Des expériences sur deux tâches de traduction automatique montrent ces modèles à
être de qualité supérieure tout en étant plus parallélisante et exigeant de manière significative
moins de temps pour s'entraîner. Notre modèle atteint 28.4 BLEU sur le WMT 2014 Anglais-
Tâche de traduction vers l'Allemagne, en améliorant par rapport aux meilleurs résultats existants, y compris
ensemble, par plus de 2 BLEU. Sur la tâche de traduction de l'anglais au français de la WMT 2014,
notre modèle établit un nouveau score BLEU à la fine pointe de la technologie de 41.8 après
formation de 3,5 jours sur huit GPU, une petite fraction des coûts de formation de la
les meilleurs modèles de la littérature. Nous montrons que le Transformateur généralise bien à
d'autres tâches en l'appliquant avec succès à la circonscription anglaise
données de formation importantes et limitées.

[1] Contribution égale. Ordre d'inscription est aléatoire. Jakob proposé de remplacer les RNN par l'auto-attention et commencé
l'effort pour évaluer cette idée. Ashish, avec Illia, a conçu et mis en œuvre les premiers modèles Transformer et
a joué un rôle crucial dans tous les aspects de ce travail.
l'attention et la représentation de la position libre de paramètres et est devenu l'autre personne impliquée dans presque chaque
Niki a conçu, implémenté, ajusté et évalué d'innombrables variantes de modèles dans notre base de codes d'origine et
tensor2tensor. Llion a également expérimenté de nouvelles variantes de modèles, a été responsable de notre base de code initiale, et
d'inférence et de visualisation efficaces. Lukasz et Aidan ont passé d'innombrables longues journées à concevoir diverses parties de et
mettre en œuvre tensor2tensor, en remplaçant notre base de code précédente, en améliorant considérablement les résultats et en accélérant massiv
nos recherches.

[†] Travail effectué pendant que chez Google Brain.
[*] Travail effectué pendant que chez Google Research.

31e Conférence sur les systèmes de traitement de l'information neuronale (NIPS 2017), Long Beach, CA, États-Unis.

1 Présentation

Réseaux neuronaux récurrents, longue mémoire à court terme [13] et récidivant fermé <a hr
en particulier, ont été fermement établis comme l'état de la technique approche dans la modélisation de séquence et
problèmes de transduction tels que la modélisation du langage et la traduction automatique <a href="attention.html#12">[35,
Depuis, les efforts ont continué de repousser les limites des modèles de langage récurrents et de l'encodeur-décodeur
architectures [38, 24, 15].

Les modèles récurrents calculent généralement les facteurs le long des positions symboliques de l'entrée et de la sortie
séquences. Alignement des positions sur les étapes du temps de calcul, elles génèrent une séquence de cache
Etats , en fonction de l'état caché précédent _ et l'entrée pour la position . Ce qui est intrinsèquement
la nature séquentielle empêche la parallélisation dans les exemples de formation, qui devient critique à plus long terme
longueurs de séquence, car les contraintes de mémoire limitent le batch à travers des exemples.
amélioration significative de l'efficacité de calcul par des astuces de factorisation [21] et cor
calcul [32], tout en améliorant la performance du modèle dans le cas de ce dernier.
Cependant, la contrainte du calcul séquentiel demeure.

Les mécanismes d'attention font désormais partie intégrante de la modélisation et de la transduction des séquences.
les modèles de travail dans diverses tâches, permettant de modéliser les dépendances sans tenir compte de leur distance en
les séquences d'entrée ou de sortie [2, 19]. Dans tous les c
sont utilisés conjointement avec un réseau récurrent.

Dans ce travail, nous proposons le Transformer, une architecture modèle évitant la récurrence et à la place
s'appuyant entièrement sur un mécanisme d'attention pour attirer les dépendances mondiales entre les intrants et les extrants.
Le Transformateur permet une parallélisation significativement plus et peut atteindre un nouvel état de l'art dans
qualité de traduction après avoir été formé pour aussi peu que douze heures sur huit P100 GPU.

2 Historique

L'objectif de réduire le calcul séquentiel forme également le fondement du GPU neuronal étendu
[16], ByteNet [18] et ConvS2S <a href="attention.html#1
bloc, calcul des représentations cachées en parallèle pour toutes les positions d'entrée et de sortie.
le nombre d'opérations nécessaires pour relier les signaux de deux positions d'entrée ou de sortie arbitraires augmente
dans la distance entre les positions, linéairement pour ConvS2S et logarithmiquement pour ByteNet.
il est plus difficile d'apprendre les dépendances entre des positions éloignées [12].
à un nombre constant d'opérations, bien qu'au prix d'une résolution efficace réduite due
à la moyenne des positions pondérées par l'attention, un effet que nous contredisons avec l'attention multi-tête comme
décrit dans la section 3.2.

L'auto-attention, parfois appelée intra-attention, est un mécanisme d'attention reliant différentes positions
d'une séquence unique afin de calculer une représentation de la séquence.
utilisé avec succès dans une variété de tâches, y compris la compréhension de la lecture, la synthèse abstraite,
représentation de la phrase indépendante de la tâche [4, 27, 28

Les réseaux de mémoire de bout en bout sont basés sur un mécanisme d'attention récurrent au lieu de séquence-
une récurrence alignée et il a été démontré que les réponses aux questions en langage simple étaient efficaces et
tâches de modélisation du langage [34].

À notre connaissance, cependant, le Transformateur est le premier modèle de transduction reposant sur
entièrement sur l'auto-attention pour calculer les représentations de son entrée et de sa sortie sans utiliser la séquence-
Dans les sections suivantes, nous allons décrire le Transformateur, motiver
auto-attention et discuter de ses avantages par rapport à des modèles tels que [17, 18] et <a h

3 Architecture du modèle

La plupart des modèles concurrentiels de transduction de séquences neurales ont une structure encodeur-décodeur <a href="at
Ici, l'encodeur cartographie une séquence d'entrée de représentations de symboles à une séquence
de représentations continues . Compte tenu de , le décodeur génère alors une sortie
séquence de symboles un élément à la fois. À chaque étape, le modèle est auto-régressif
[10], consommant les symboles précédemment générés comme entrée supplémentaire lors d

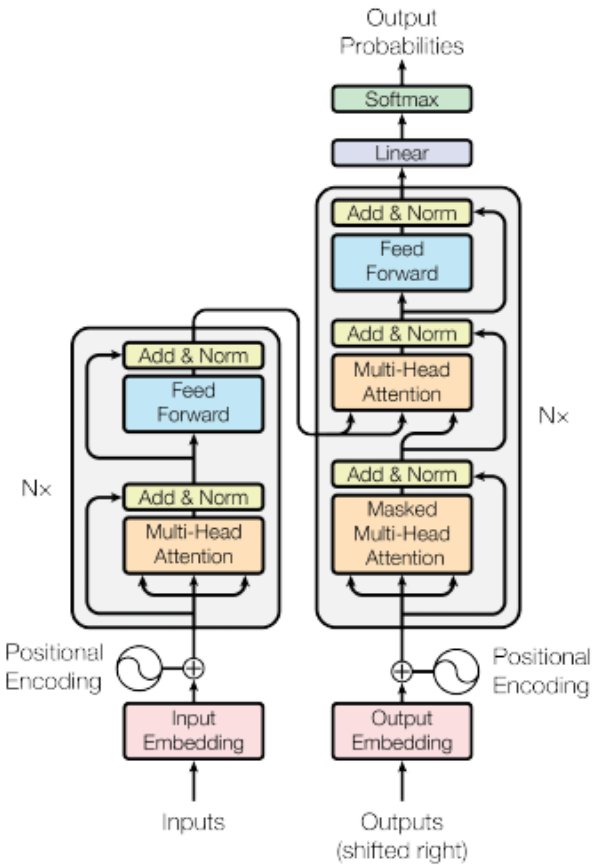


Figure 1: Le transformateur - architecture de modèle.

Le Transformateur suit cette architecture globale en utilisant l'auto-attention empilée et pointue, entièrement couches connectées pour l'encodeur et le décodeur, montrées dans les moitiés gauche et droite de la figure <a href="attention.l respectivement.

3.1 Le Procès-verbal de la séance du [illegible] entre en vigueur le vingtième jour suivant celui de sa publication au Journal officiel de l'Unio

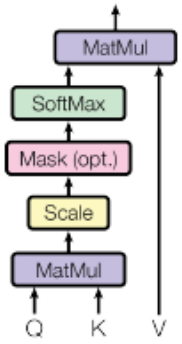
Encodeur & L'encodeur est composé d'une pile de couches identiques. Chaque couche a deux Les sous-couches. La première est un mécanisme d'auto-attention multi-têtes, et la seconde est une simple, position- Nous utilisons une connexion résiduelle [11] autour de chacun des les deux sous-couches, suivies de la normalisation des couches [1]. C'est-à-dire que la sortie , où est la fonction implémentée par la sous-couche Pour faciliter ces connexions résiduelles, toutes les sous-couches du modèle, ainsi que l'intégration couches, produire des sorties de dimension C'est vrai.

Décoder : Le décodeur est également composé d'une pile de couches identiques. En plus des deux sous-couches dans chaque couche d'encodeur, le décodeur insère une troisième sous-couche, qui effectue plusieurs têtes attention sur la sortie de la pile d'encodeur. Similaire à l'encodeur, nous employons des connexions résiduelles autour de chacune des sous-couches, suivie d'une normalisation des couches. Nous modifions également l'auto-attention sous-couche dans la pile de décodeurs pour empêcher les positions de s'occuper de positions ultérieures. masquer, combiné avec le fait que les intégrations de sortie sont compensées par une position, assure que le prévisions pour la position ne peut dépendre que des sorties connues à des positions inférieures à C'est vrai.

3.2. À l ' attention du Comité

Une fonction d'attention peut être décrite comme mappage d'une requête et d'un ensemble de paires de valeurs clés à une sortie où la requête, les clés, les valeurs et la sortie sont tous des vecteurs. La sortie est calculée comme une somme pondérée

Attention au produit à point d'échelle



Attention multi-têtes

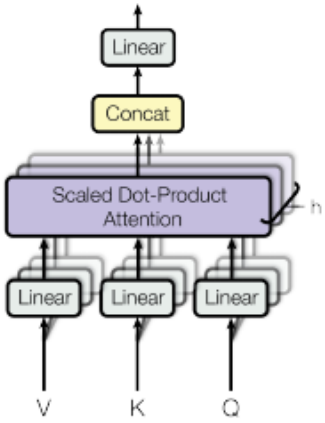


Figure 2 : (à gauche) Attention à l'échelle du produit à point. (à droite) L'attention multi-têtes se compose de plusieurs couches d'attention fonctionnant en parallèle.

des valeurs, lorsque le poids attribué à chaque valeur est calculé par une fonction de compatibilité de la requête avec la clé correspondante.

3.2.1 Attention au produit à point d'échelle

Nous attirons notre attention particulière sur « l'attention sur les produits à points calibrés » (figure <a href="attention.html#4" requêtes et clés de dimension , et valeurs de dimension . Nous calculons les produits point de la requête avec toutes les clés, diviser chacun par et appliquer une fonction softmax pour obtenir les poids sur le valeurs.

Dans la pratique, nous calculons la fonction d'attention sur un ensemble de requêtes simultanément, emballées ensemble dans une matrice. Les clés et les valeurs sont également emballées ensemble dans des matrices et . Nous calculons la matrice des résultats comme suit:

(1)

Les deux fonctions d'attention les plus couramment utilisées sont l'attention additive [2], et l'attention plicative. L'attention du produit dot est identique à notre algorithme, à l'exception du facteur d'échelle des . L'attention additive calcule la fonction de compatibilité à l'aide d'un réseau vers l'avant avec Une seule couche cachée. Alors que les deux sont similaires dans la complexité théorique, l'attention du produit de point est beaucoup plus rapide et plus efficace de l'espace dans la pratique, car il peut être mis en œuvre en utilisant hautement optimisé code de multiplication matricielle.

Alors que pour de petites valeurs des deux mécanismes fonctionnent de la même manière, l'attention additive surpasse les performa dot attention produit sans mise à l'échelle pour des valeurs plus grandes de l'attention.html#10">[3]. Nous soupçonnons que pou , les produits à points grandissent, poussant la fonction softmax dans les régions où il a gradients extrêmement petits. Pour contrer cet effet, nous C'est alors les produits point par

3.2.2. Attention multi-têtes

Au lieu d'effectuer une seule fonction d'attention avec -les clés, valeurs et requêtes dimensionnelles, nous avons trouvé utile de projeter linéairement les requêtes, les clés et les valeurs fois avec différents, appris projections linéaires , et sur chacune de ces versions projetées de requêtes, clés et valeurs nous effectuons ensuite la fonction d'attention en parallèle, donnant dimensionnel

Pour illustrer pourquoi les produits à points deviennent grands, supposons que les composantes de sont indépendantes variables avec moyenne et de la variation Puis leur produit de point, , a moyen et de la variation C'est vrai.

valeurs de sortie. Ces valeurs sont concaténées et une fois de plus projetées, ce qui donne les valeurs finales, comme représenté dans la figure 2.

L'attention multi-têtes permet au modèle de s'occuper conjointement de l'information provenant de différentes représentations Avec une seule tête d'attention, la moyenne inhibe cela.

où

Lorsque les projections sont des matrices de paramètres ;
et C'est vrai.

Dans ce travail, nous employons couches d'attention parallèles, ou têtes. Pour chacun de ces
. En raison de la dimension réduite de chaque tête, le coût total de calcul
est similaire à celui de l'attention d'une seule tête avec la pleine dimensionnalité.

3.2.3 Applications de l'attention dans notre modèle

Le Transformateur utilise l'attention multi-têtes de trois façons différentes:

- Dans les calques "encoder-décoder attention", les requêtes proviennent du calque de décodeur précédent, et les clés et valeurs de mémoire proviennent de la sortie de l'encodeur.
  position dans le décodeur pour assister à toutes les positions dans la séquence d'entrée.
  mécanismes d'attention encodeur-décodeur typiques dans les modèles séquence à séquence tels que [38, 2, 9].
- L'encodeur contient des couches d'auto-attention. Dans une couche d'auto-attention toutes les clés, valeurs et les requêtes viennent du même endroit, dans ce cas, la sortie de la couche précédente dans le encodeur. Chaque position de l'encodeur peut s'occuper de toutes les positions de la couche précédente de l'encodeur. Encodeur.
- De même, les couches d'auto-attention dans le décodeur permettent à chaque position du décodeur de s'occuper de toutes les positions dans le décodeur jusqu'à et y compris cette position. Nous devons empêcher vers la gauche flux d'information dans le décodeur pour préserver la propriété auto-régressive. Nous implémentons ceci à l'intérieur de l'attention à l'échelle du produit d'un point en masquant) toutes les valeurs dans l'entrée
  Voir la figure 2.

3.3. Réseaux d'alimentation orientés vers la position

En plus des sous-couches d'attention, chacune des couches de notre codeur et décodeur contient un réseau d'alimentation connecté vers l'avant, qui est appliqué à chaque position séparément et de manière identique. se compose de deux transformations linéaires avec une activation ReLU entre les deux.

(2)

Bien que les transformations linéaires soient les mêmes pour différentes positions, elles utilisent des paramètres différents.
Une autre façon de décrire ceci est comme deux convolutions avec la taille du noyau 1.
La dimensionnalité de l'entrée et de la sortie est , et la couche intérieure a dimensionnalité
C'est vrai.

3,4 Embeddings et Softmax

De même que d'autres modèles de transduction de séquence, nous utilisons des incrustations apprises pour convertir l'entrée jetons et jetons de sortie vers des vecteurs de dimension . Nous utilisons également la transfor- linéaire apprise habituelle mation et fonction softmax pour convertir la sortie du décodeur en probabilités prédites à la suite.
notre modèle, nous partageons la même matrice de poids entre les deux couches d'intégration et le pré-softmax transformation linéaire, semblable à [30]. Dans les couches d'intégration, nous multiplions c
C'est vrai.

Tableau 1: Longueurs maximales du trajet, complexité par couche et nombre minimal d'opérations séquentielles pour différents types de couches. est la longueur de la séquence, est la dimension de représentation, est le noyau taille des convolutions et la taille du quartier dans l'auto-attention restreinte.

Type de calque	Complexité par couche	Séquentiel Opérations	Longueur maximale du sentier
L'auto-attention			
Récidive			
Convolutionnel			
Auto-attention (restriction)			

3,5 Encodage positionnel

Étant donné que notre modèle ne contient aucune récurrence et aucune convolution, afin que le modèle fasse usage de la ordre de la séquence, nous devons injecter quelques informations sur la position relative ou absolue de la Jetons dans la séquence. À cette fin, nous ajoutons des "encodages positionnels" à l'encodage d'entrée à la les fonds des piles d'encodeur et de décodeur. Les encodages positionnels ont la même dimension

comme les encodages, de sorte que les deux peuvent être résumés. Il y a beaucoup de choix d'encodages positionnels, appris et corrigé [9].

Dans ce travail, nous utilisons des fonctions sinus et cosinus de fréquences différentes:

où est la position et est la dimension. Autrement dit, chaque dimension de l'encodage positionnel correspond à un sinusoïde. Les longueurs d'onde forment une progression géométrique de à . Nous avons a choisi cette fonction parce que nous avons émis l'hypothèse qu'elle permettrait au modèle d'apprendre facilement à assister à postes relatifs, depuis pour tout décalage fixe , peut être représenté comme une fonction linéaire de C'est vrai.

Nous avons également expérimenté l'utilisation d'intégrations positionnelles apprises [9] en les versions ont produit des résultats presque identiques (voir Tableau 3 row (E)). Nous avons parce qu'il peut permettre au modèle d'extrapoler à des longueurs de séquence plus longues que celles rencontrées pendant l'entraînement.

4 Pourquoi l'auto-attention

Dans cette section, nous comparons divers aspects des couches d'auto-attention à la récurrence et à la convolu- les couches nationales couramment utilisées pour la cartographie d'une séquence de symboles de longueur variable à une autre séquence de longueur égale , avec , comme un caché couche dans un encodeur de transduction de séquence typique ou décodeur. Motivant notre utilisation de l'auto-attention nous Considérez trois desiderata.

Un est la complexité totale du calcul par couche. Un autre est la quantité de calcul qui peut être parallélisés, tels que mesurés par le nombre minimal d'opérations séquentielles requis.

Le troisième est la longueur du chemin entre les dépendances à long terme dans le réseau. les dépendances sont un défi clé dans de nombreuses tâches de transduction de séquence. la capacité d'apprendre de telles dépendances est la longueur des voies vers l'avant et vers l'arrière des signaux doivent traverser dans le réseau. Plus ces chemins sont courts entre n'importe quelle combinaison de positions dans l'entrée et les séquences de sortie, plus il est facile d'apprendre les dépendances à longue distance <a href="attention.html#11">[12]. < la longueur maximale du chemin entre deux positions d'entrée et de sortie dans les réseaux composés de différents types de couches.

Comme indiqué dans le tableau 1, une couche d'auto-attention relie toutes les positions avec u opérations exécutées, alors qu'une couche récurrente les opérations séquentielles. complexité informatique, les couches d'auto-attention sont plus rapides que les couches récurrentes lorsque la séquence

longueur est plus petit que la dimension de représentation , ce qui est le plus souvent le cas avec
représentations de phrases utilisées par des modèles à la fine pointe de la technologie dans les traductions de machines, comme
[38] et byte-pair [31] représentations.
très longues séquences, l'auto-attention pourrait être limitée à considérer seulement un quartier de taille
dans
la séquence d'entrée centrée autour de la position de sortie respective. Cela augmenterait le maximum
longueur du chemin jusqu'à . Nous envisageons d'approfondir cette approche dans les travaux futurs.

Une seule couche convolutionnelle avec largeur du noyau ne connecte pas toutes les paires d'entrée et de sortie
Le fait de le faire nécessite une pile de couches convolutionnelles dans le cas de grains contigus,
ou dans le cas de convolutions dilatées [18], en augmentant la longueur des che
entre deux positions dans le réseau. Les couches convolutionnelles sont généralement plus chères que
les couches récurrentes, par un facteur de . Convolutions séparables [6], toutefois, diminuer la comp
considérablement, pour . Même avec , cependant, la complexité d'un
la convolution est égale à la combinaison d'une couche d'auto-attention et d'une couche d'alimentation ponctuelle,
l'approche que nous adoptons dans notre modèle.

Comme avantage secondaire, l'auto-attention pourrait donner des modèles plus interprétables. Nous inspectons les distribution
de nos modèles et de présenter et de discuter des exemples dans l'annexe. Non seulement l'attention individuelle
têtes clairement apprendre à effectuer des tâches différentes, beaucoup semblent présenter un comportement lié à la syntaxique
et la structure sémantique des phrases.

5 Formation

Cette section décrit le régime de formation de nos modèles.

5.1. Données sur la formation et l'abattement

Nous avons suivi une formation sur l'ensemble de données anglais-allemand standard WMT 2014 qui se compose d'environ 4,
les paires de phrases. Les phrases ont été encodées en utilisant l'encodage octet-pair [3], qui a
vocabulaire cible d'environ 37000 jetons. Pour le français-anglais, nous avons utilisé le WMT significativement plus grand
Ensemble de données français-anglais 2014 composé de 36 millions de phrases et de jetons divisés en 32000 mots-clés
vocabulaire [38]. Les paires de phrases ont été groupées par la longueur approximative de la
lot contenait un ensemble de paires de phrases contenant environ 25000 jetons source et 25000
des jetons cibles.

5.2. Matériel et calendrier

Nous avons formé nos modèles sur une seule machine avec 8 processeurs NVIDIA P100.
les hyperparamètres décrits dans l'article, chaque étape d'entraînement a pris environ 0,4 seconde.
a formé les modèles de base pour un total de 100 000 étapes ou 12 heures. Pour nos grands modèles, (décrit sur le
bas de la ligne de tableau 3), le temps d'étape était de 1,0 seconde. Les grands modèles ont été
(3,5 jours).

5,3 Optimisateur

Nous avons utilisé l'optimiseur Adam [20] . Nous avons varié l'apprentissage
taux sur le cours de formation, selon la formule:

(3)

Cela correspond à une augmentation linéaire du taux d'apprentissage pour la première des étapes de formation,
et la diminuer par la suite proportionnellement à la racine carrée inverse du nombre d'étapes.
C'est vrai.

5,4 Régularisation

Nous employons trois types de régularisation au cours de la formation :

Tableau 2: Le transformateur obtient de meilleurs scores de l'UEBL que les modèles de pointe précédents sur le Test de nouvelles de l'anglais vers l'allemand et de l'anglais vers le français2014 à une fraction du coût de la formation.

Modèle	UEBL		Coût de la formation (POFT)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		**39.2**		
GNMT + RL [38]	24.6	39.92		
ConvS2S [9]	25.16	40.46		
Ministère de l'éducation [32]	26.03	40.56		
Ensemble Deep-Att + PosUnk [39]		40.4		
GNMT + RL Ensemble [38]	26.30	41.16		
Ensemble ConvS2S [9]	26.36	41.29		
Transformateur (modèle de base)	27,3	38,1		
Transformateur (grand)	**28.4**	**41,8**		

Décrochage résiduel Nous appliquons l'abandon [33] à la sortie de chaque sous-couche, ava
En outre, nous appliquons l'abandon scolaire aux sommes des emboîtements et des
les encodages positionnels dans les piles d'encodeur et de décodeur. Pour le modèle de base, nous utilisons un taux de
C'est vrai.

Lissage des étiquettes Au cours de la formation, nous avons employé le lissage des étiquettes de valeur <a href="attention.html#12">[36]. </
Bless perplexité, comme le modèle apprend à être plus incertain, mais améliore la précision et le score BLEU.

6 Résultats

6.1. Traduction automatique

Sur la tâche de traduction de l'anglais à l'allemand WMT 2014, le grand modèle de transformateur (transformateur (grand)
dans le tableau 2) performe les meilleurs modèles déjà signalés (y compris les ensembles) de
L'UEBL, établissant un nouveau score de l'UEBL à la fine pointe de la technologie. Le configuration de ce modèle est
liste dans le bas de la ligne du tableau 3. Formation prise jours sur 8 GPU. Même notre modèle de base
surpasse tous les modèles et ensembles publiés précédemment, à une fraction du coût de la formation de l'un ou l'autre des
les modèles compétitifs.

Sur la tâche de traduction de l'anglais au français de la WMT 2014, notre grand modèle obtient un score BLEU de
surperformant tous les modèles uniques publiés précédemment, à moins de le coût de la formation
précédent modèle de pointe. Le modèle Transformer (grand) formé pour l'anglais au français utilisé
Taux d ' abandon scolaire, au lieu de C'est vrai.

Pour les modèles de base, nous avons utilisé un modèle unique obtenu en calculant la moyenne des 5 derniers points de contrô
Pour les grands modèles, nous avons effectué en moyenne les 20 derniers points de contrôle.
recherche de faisceau utilisé avec une taille de faisceau de le taux de la peine [38]. Ces hyperparam
ont été choisis après expérimentation sur l'ensemble de développement. Nous avons défini la longueur de sortie maximale pen
Inférence à la longueur d'entrée, mais se terminent tôt si possible [38].

Tableau 2 résume nos résultats et compare nos coûts de traduction et de formation à d'autres
Nous estimons le nombre d'opérations flottantes utilisées pour former un
modèle en multipliant le temps de formation, le nombre de GPU utilisés, et une estimation du
Capacité à point flottant d'une seule précision de chaque GPU .

6.2 Variations du modèle

Pour évaluer l'importance des différentes composantes du Transformateur, nous avons varié notre modèle de base.
de différentes manières, en mesurant le changement de performance sur la traduction de l'anglais à l'allemand sur le

Nous avons utilisé des valeurs de 2,8, 3,7, 6,0 et 9,5 TFLOPS pour K80, K40, M40 et P100, respectivement.

Tableau 3: Variations sur l'architecture Transformer. Les valeurs non listées sont identiques à celles de la base modèle. Toutes les mesures sont sur l'ensemble de développement de traduction de l'anglais à l'allemand, newstest2013. les perplexités sont par mot-clé, selon notre encodage byte-pair, et ne devraient pas être comparées à perplexités par mot.

									train étapes	PPL (dev)	UEBL (dev)	Params
de base	6	512	2048	8	64	64	0,1	0,1	100K	4.92	25.8	65
(A)				1	512	512				5.29	24,9	
				4	128	128				5,00	25.5	
				16	32 ans	32 plus				4.91	25.8	
				32 ans	16	16				5.01	25.4	
(B)					16					5.16	25.1	58
					32 ans et plus					5.01	25.4	60
(C)	2									6.11	23.7	36
	4									5.19	25,3	50 ans
	8									4.88	25.5	Quatre-vingt-dix
		252	000	000	000	000	32 ans	32 plus	et plus	5.75	24.5	Les États-Unis d'Amérique
		1024				128	128			4.66	26,0	168
			1024							5.12	Le présent règlement entre en vigueur le vingti	53
			4096							4.75	26.2	90 États-Unis d'Amérique
(D)							0	0	0	5.77	24,6	0
							0,2			4.95	25.5	
								0	0	4.67	25,3	0
								0,2		5.47	25.7	
(E)					emboîtement positionnel au lieu des sinusoïdes					4.92	25.7	
gros	6	1024	4096	16			0,3		300K	**4,33**	**26,4**	213

ensemble de développement, newstest2013. Nous avons utilisé la recherche de faisceau comme décrit dans la section précéden
Nous présentons ces résultats dans le tableau 3.

Dans le tableau 3 rows (A), nous varions le nombre de têtes d'attention et les dimensions de l
le maintien de la quantité de constante de calcul, tel que décrit dans la section 3.2.2.
L'attention est de 0,9 BLEU pire que le meilleur réglage, la qualité baisse également avec trop de têtes.

Dans le tableau 3 rows (B), nous observons que la réduction de la taille de la clé d'attention
suggère que la détermination de la compatibilité n'est pas facile et qu'une compatibilité plus sophistiquée
Nous observons en outre dans les lignes (C) et (D) que, comme prévu,
Les modèles plus grands sont meilleurs, et l'abandon est très utile pour éviter le sur-ajustement.
Encodage positionnel sinusoïdal avec des encodages positionnels apprises [9], et observer p
résultats au modèle de base.

6,3 Parsing de la circonscription d'anglais

Pour évaluer si le Transformateur peut généraliser à d'autres tâches nous avons effectué des expériences sur l'anglais
Cette tâche présente des défis spécifiques: la production est soumise à des contraintes structurelles fortes.
et est significativement plus long que l'entrée. De plus, la séquence RNN à la séquence
les modèles n'ont pas été en mesure d'atteindre des résultats de pointe dans les régimes de petites données <a href="attention.h

Nous avons formé un transformateur 4 couches avec dans la partie du Wall Street Journal (WSJ)
Penn Treebank [25], environ 40K phrases d'entraînement. Nous l'avons également formé da
utilisant le plus grand corps de haute confiance et BerkleyParser avec des phrases d'environ 17M
[37]. Nous avons utilisé un vocabulaire de jetons 16K pour le réglage WSJ seulement et un
pour le réglage semi-supervisé.

Nous n'avons effectué qu'un petit nombre d'expériences pour sélectionner l'abandon, à la fois l'attention et les résidus.
(section 5.4), taux d'apprentissage et taille du faisceau sur l'ensemble de développement de la s
Nous sommes restés inchangés par rapport au modèle de traduction de base anglais-allemand.

Tableau 4 : Le transformateur généralise bien l'analyse des circonscriptions anglaises (les résultats sont donnés à l'article 23) de WSJ)

Analyseur	Formation	WSJ 23 F1
Vinyals & Kaiser el al. (2014) [37]	WSJ seulement, discriminant	88,3
Petrov et al. (2006) [29]	WSJ seulement, discriminant	90,4
Zhu et al. (2013) [40]	WSJ seulement, discriminant	90,4
Dyer et al. (2016) [8]	WSJ seulement, discriminant	91,7
Transformateur (4 couches)	WSJ seulement, discriminant	91,3
Zhu et al. (2013) [40]	semi-supervisé	91,3
Huang & Harper (2009) [14]	semi-supervisé	91,3
McClosky et al. (2006) [26]	semi-supervisé	92,1
Vinyals & Kaiser el al. (2014) [37]	semi-supervisé	92,1
Transformateur (4 couches)	semi-supervisé	92,7
Luong et al. (2015) [23]	multi-tâches	93,0
Dyer et al. (2016) [8]	génératif	93.3

augmentation de la longueur maximale de sortie à la longueur d'entrée + Nous avons utilisé une taille de faisceau de pour WSJ seulement et le réglage semi-supervisé.

Nos résultats dans le tableau 4 montrer que malgré l'absence d'accord spécifique à la tâche n Bien sûr, ce qui donne de meilleurs résultats que tous les modèles précédemment mentionnés, à l'exception de Grammaire du réseau neuronal récurrent [8].

Contrairement aux modèles de séquence à séquence RNN [37], le Transformateur surpasse l Parser [29] même lorsque l'entraînement n'est effectué que sur l'ensemble d'entraînement W

7 Le présent règlement entre en vigueur le vingtième jour suivant celui de sa publication au Jourr

Dans ce travail, nous avons présenté le Transformer, le premier modèle de transduction de séquence basé entièrement sur attention, en remplaçant les couches récurrentes les plus couramment utilisées dans les architectures encoder-decoder par l'auto-attention multi-tête.

Pour les tâches de traduction, le Transformateur peut être formé beaucoup plus rapidement que les architectures basées sur sur les couches récurrentes ou convolutionnelles. Sur WMT 2014 Anglais-Allemand et WMT 2014 Tâches de traduction de l'anglais au français, nous atteignons un nouvel état de l'art. Dans l'ancienne tâche, notre meilleur le modèle surpasse même tous les ensembles précédemment signalés.

Nous sommes ravis de l'avenir des modèles axés sur l'attention et nous prévoyons de les appliquer à d'autres tâches. projeter d'étendre le transformateur aux problèmes concernant les modalités d'entrée et de sortie autres que le texte et d ' enquêter sur les mécanismes locaux d ' attention limitée permettant de traiter efficacement les gros intrants et produits Comme les images, l'audio et la vidéo, rendre la génération moins séquentielle est un autre objectif de recherche de notre part.

Le code que nous avons utilisé pour former et évaluer nos modèles est disponible à l'adresse suivante https://github.com/tensorflow/tensor2tensor

Remerciements Nous sommes reconnaissants à Nal Kalchbrenner et Stephan Gouws pour leur contribution fructueuse. commentaires, corrections et inspiration.

Références

[1] Jimmy Lei Ba, Jamie Ryan Kiros et Geoffrey E Hinton. Normalisation des couches. , 2016.

[2] Dzmitry Bahdanau, Kyunghyun Cho et Yoshua Bengio. Traduction automatique neuronale en commun apprendre à s'aligner et à traduire. , abs/1409.0473, 2014.

[3] Denny Britz, Anna Goldie, Minh-Thang Luong et Quoc V. Le. Exploration massive de neurones architectures de traduction automatique. , abs/1703.03906, 2017.

[4] Jianpeng Cheng, Li Dong et Mirella Lapata. Longs réseaux de mémoire à court terme pour machine lecture. , 2016.

[5] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, et Yoshua Bengio. Représentations de phrases d'apprentissage utilisant rnn encoder-decoder pour les statistiques traduction automatique. , abs/1406.1078, 2014.

[6] Francois Chollet. Xception: Apprentissage profond avec des convolutions séparables en profondeur. , 2016.

[7] Junyoung Chung, Caglar Gülçehre, Kyunghyun Cho et Yoshua Bengio. Évaluation empirique des réseaux neuraux récurrents fermés sur la modélisation des séquences. abs/1412.3555, 2014.

[8] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros et Noah A. Smith. Neural récurrent grammaires réseau. , 2016.

[9] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats et Yann N. Dauphin. l'apprentissage séquentiel. , 2017.

[10] Alex Graves. Générer des séquences avec des réseaux neuronaux récurrents. , 2013.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren et Jian Sun. la reconnaissance de l'âge. , pages 770 à 778, 2016.

[12] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi et Jürgen Schmidhuber. les filets récurrents: la difficulté d'apprendre les dépendances à long terme, 2001.

[13] Sepp Hochreiter et Jürgen Schmidhuber. Longue mémoire à court terme. , 9(8):1735–1780, 1997.

[14] Zhongqiang Huang et Mary Harper. Grammaires PCFG auto-entraînement avec annotations latentes dans toutes les langues. , pages 832 à 841. ACL, août 2009.

[15] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer et Yonghui Wu. Exploration les limites de la modélisation des langues. , 2016

[16] Łukasz Kaiser et Samy Bengio. La mémoire active peut-elle remplacer l'attention? , 2016.

[17] Łukasz Kaiser et Ilya Sutskever. Les GPU neuronaux apprennent les algorithmes. , 2016.

[18] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves et Koray Kavukcuoglu. Traduction automatique neuronale dans le temps linéaire. , Pour l'année 2017.

[19] Yoon Kim, Carl Denton, Luong Hoang et Alexander M. Rush. Réseaux d'attention structurés. Dans , 2017.

[20] Diederik Kingma et Jimmy Ba. Adam: Une méthode pour l'optimisation stochastique. , 2015.

[21] Oleksii Kuchaiev et Boris Ginsburg. Astuces de factorisation pour les réseaux LSTM. , 2017.

[22] Zhouhan Lin, Minwei Feng, Cicéron Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, et Yoshua Bengio. Une phrase d'auto-attention structurée encastrant. , 2017.

[23] Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, et Lukasz Kaiser. séquence pour séquencer l'apprentissage. , 2015.

[24] Minh-Thang Luong, Hieu Pham et Christopher D Manning. traduction automatique neurale basée. , 201

[25] Mitchell P Marcus, Mary Ann Marcinkiewicz, et Beatrice Santorin. Construction d'un grand annoté corpus de l'anglais: La banque d'arbres de penn. , 19(2):313-330, 1993.

[26] David McClosky, Eugene Charniak et Mark Johnson. Autoformation efficace pour l'analyse. , ACL, juin 2006.

[27] Ankur Parikh, Oscar Täckström, Dipanjan Das et Jakob Uszkoreit. Une attention décomposable modèle. , 2016.

[28] Romain Paulus, Caiming Xiong et Richard Socher. Un modèle profondément renforcé pour l'abstraction une synthèse. , 2017.

[29] Slav Petrov, Leon Barrett, Romain Thibaux et Dan Klein. Apprentissage précis, compact, et l'annotation d'arbre interprétable. , pages 433 à 440. ACL, juillet 2006

[30] Ofir Press et Lior Wolf. Utilisation de l'intégration de sortie pour améliorer les modèles de langage. , 2016.

[31] Rico Sennrich, Barry Haddow et Alexandra Birch. Traduction automatique neurale de mots rares avec des unités de sous-mots. , 2015.

[32] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, et Jeff Dean. Des réseaux neuronaux de grande envergure : le mélange d'experts éparpillé couche. , 2017.

[33] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever et Ruslan Salakhutdinov. Dropout : un moyen simple d'éviter que les réseaux neuraux ne s'adaptent trop. , 15(1):1929–1958, 2014.

[34] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston et Rob Fergus. Mémoire de bout en bout Dans C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama et R. Garnett, rédacteurs en chef, , pages 2440–2448. Curran Associates, Inc., 2015.

[35] Ilya Sutskever, Oriol Vinyals, et Quoc VV Le. Séquence pour l'apprentissage de séquence avec neurone les réseaux. , pages 3104–3112, 2014.

[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens et Zbigniew Wojna. Repenser l'architecture de départ pour la vision informatique. , abs/1512.00567, 2015.

[37] Vinyals & Kaiser, Koo, Petrov, Sutskever et Hinton. Grammaire comme langue étrangère. , 2015.

[38] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. système de traduction: Combler l'écart entre la traduction humaine et la traduction automatique. , 2016.

[39] Jie Zhou, Ying Cao, Xuguang Wang, Peng Li et Wei Xu. connexions rapides pour la traduction automatique neuronale. , abs/1606.04199, 2016.

[40] Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang et Jingbo Zhu. Rapide et précis Paramétrage des constituants de la réduction des déplacements. , pages 434 à 443. ACL, août 2013.

Visualisations de l'attention

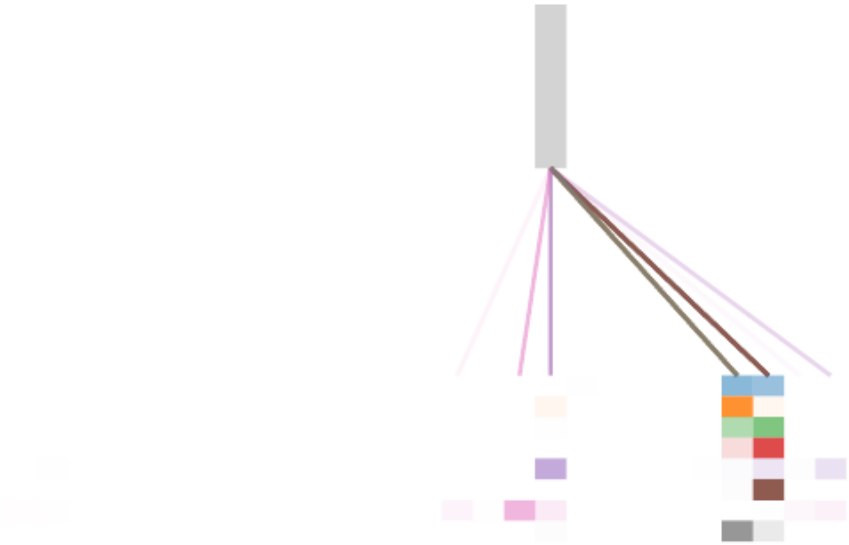


Figure 3: Un exemple du mécanisme d'attention à la suite de dépendances interurbaines
Encoder auto-attention dans la couche 5 de 6. Beaucoup des têtes d'attention s'occupent d'une dépendance lointaine de le verbe ‘faire, compléter la phrase ‘faire...plus difficile.
le mot ‘making. Différentes couleurs représentent des têtes différentes. Meilleur vu en couleur.

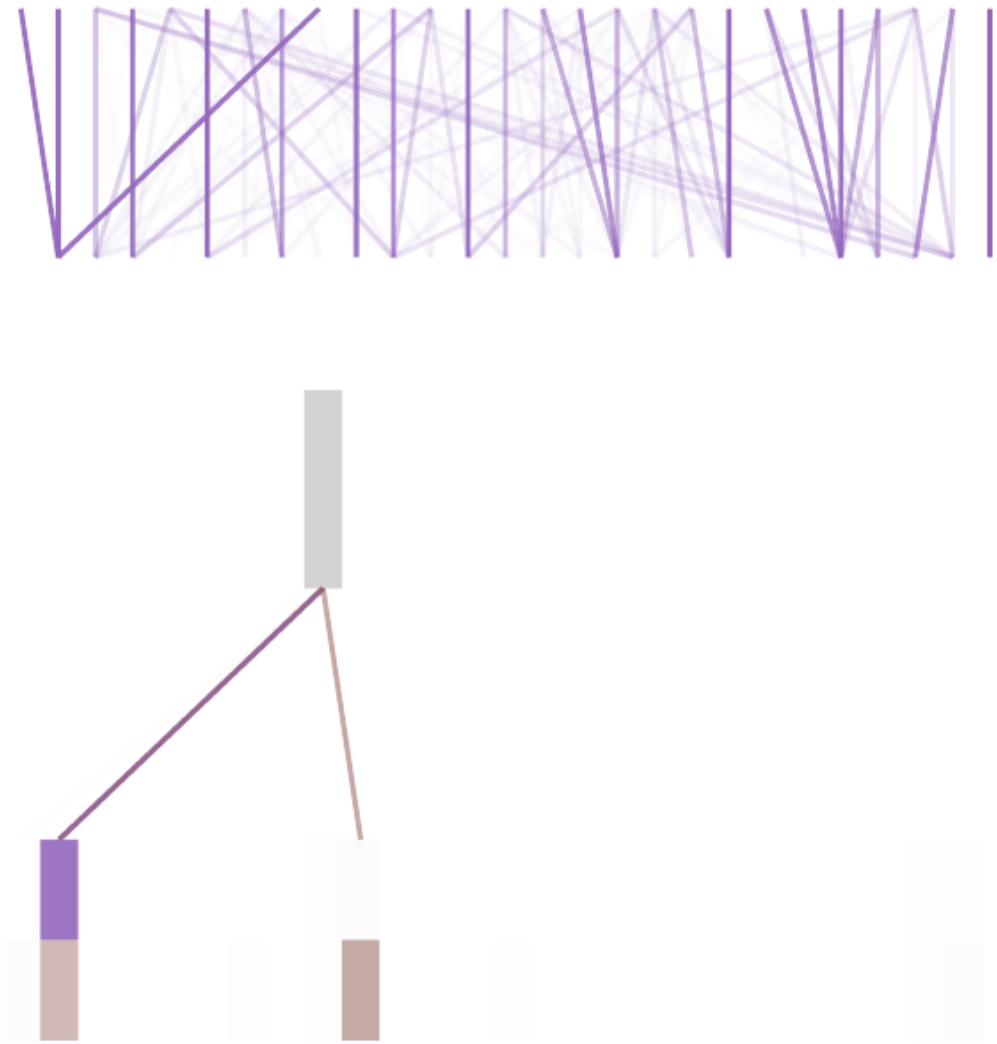


Figure 4 : Deux têtes d'attention, également dans la couche 5 de 6, apparemment impliquées dans la résolution de l'anaphore. Pleines attentions pour la tête 5. Bottom: Des attentions isolées de juste le mot «sit» pour les têtes d'attention 5
6. Notez que les attentions sont très nettes pour ce mot.

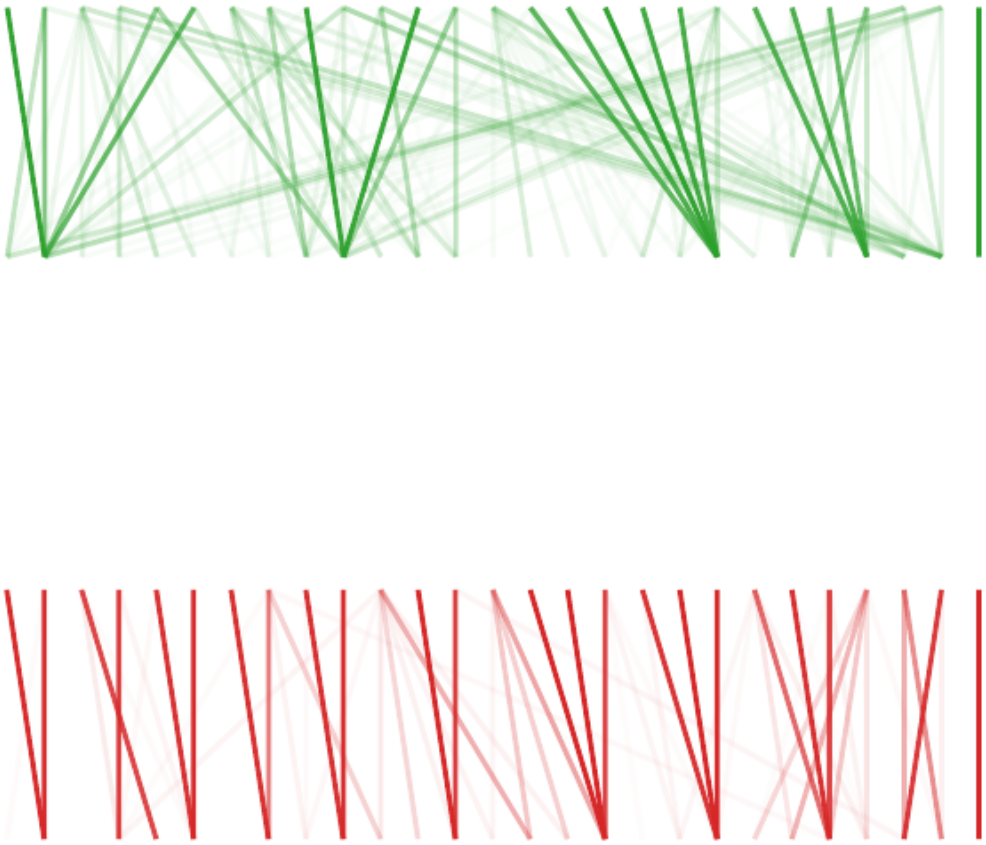


Figure 5: Nombre des têtes d'attention présentent un comportement qui semble lié à la structure de la phrase. Nous donnons deux exemples de ce type ci-dessus, à partir de deux têtes différentes de l'auto-attention encodeur à la couche 5 de 6. Les têtes ont clairement appris à accomplir différentes tâches.