

Attention Is All You Need

Synthèse en français du papier qui a donné le Transformer — ce qu'il dit exactement, pourquoi il a tout emporté, et ce qu'il faut en retenir neuf ans plus tard.

Papier — Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin.

« Attention Is All You Need », *Advances in Neural Information Processing Systems 30* (NIPS 2017). Prépublication : arXiv:1706.03762, déposée le 12 juin 2017.

Synthèse — Labo LLM, 6 septembre 2026.

Certification — Sourced littérature. Tous les chiffres cités proviennent du papier lui-même ; les rapprochements avec l'état de l'art actuel sont signalés comme tels.

En bref

En juin 2017, huit chercheurs de Google publient un papier de douze pages sur la traduction automatique. Le titre est une provocation : *l'attention est tout ce dont vous avez besoin*. Le message tient en une phrase — on peut retirer d'un modèle de séquence tout ce qui faisait sa structure depuis vingt ans, la récurrence et la convolution, et ne garder qu'un seul mécanisme, l'attention. Le résultat s'appelle le **Transformer**.

Les chiffres annoncés dans le résumé du papier : 28,4 BLEU sur la traduction anglais-allemand de WMT 2014, soit plus de 2 points au-dessus du meilleur résultat existant, ensembles de modèles compris ; 41,8 BLEU sur l'anglais-français, nouveau record pour un modèle unique, obtenu après **trois jours et demi d'entraînement sur huit GPU** — une fraction du coût des meilleurs modèles de la littérature.

Ce qui a suivi dépasse largement la traduction. L'architecture décrite dans ce papier est celle de GPT, de Claude, de Gemini, de Llama, de Mistral. Aucun des grands modèles de langage d'aujourd'hui ne s'en écarte fondamentalement. Cette synthèse explique ce que le papier dit réellement — pas ce qu'on lui prête.

EN CLAIR

Avant 2017, pour lire une phrase, un modèle la parcourait mot après mot, comme on déroule une bande magnétique : impossible d'aller au mot 40 sans avoir traversé les 39 précédents. Le Transformer remplace la bande par une table : tous les mots sont posés à plat, chacun regarde tous les autres d'un coup. C'est ce changement — de la file d'attente à la table ronde — qui a tout débloqué.

Pourquoi ce texte est fondateur

Il faut se remettre dans l'état du domaine en juin 2017. Le traitement du langage vit alors sous le régime des réseaux récurrents depuis près de vingt ans. On sait faire de la traduction correcte, mais chaque progrès coûte cher : les modèles s'entraînent pendant des semaines, ils butent sur les phrases longues, et surtout ils **ne savent pas profiter du matériel**. Les GPU s'étaient couverts de milliers de cœurs capables de travailler en même temps ; la récurrence, elle, impose de traiter les mots l'un après l'autre. On achetait de la puissance parallèle pour faire tourner un calcul qui refuse d'être parallélisé.

Le papier lève ce verrou. Et c'est là que se joue son importance réelle : sa contribution majeure n'est pas d'avoir gagné deux points de BLEU, c'est d'avoir rendu l'entraînement **parallélisable**. À partir du moment où la longueur de la phrase ne dicte plus le nombre d'étapes séquentielles, on peut jeter des machines sur le problème. Le facteur limitant cesse d'être le temps, il devient le budget.

Cette bascule a une conséquence que personne n'annonce dans le papier, et qui va pourtant tout décider : elle rend possible le **passage à l'échelle**. Une architecture qui absorbe autant de calcul qu'on lui en donne peut être agrandie, et on peut mesurer ce que l'agrandissement rapporte. Les travaux suivants n'inventeront pas une nouvelle architecture — ils prendront celle-ci et la feront grossir.

La suite tient en quelques dates. En 2018, deux équipes exploitent chacune une moitié de l'architecture : l'encodeur seul devient BERT, qui balaye les records de compréhension du langage ; le décodeur seul devient GPT, qui apprend à écrire en prédisant le mot suivant. En 2020, GPT-3 montre qu'à cette échelle, un modèle entraîné à cette seule tâche se met à accomplir des tâches qu'on ne lui a jamais enseignées. Tout ce qui suit — les assistants conversationnels, les modèles de code, les modèles multimodaux — reste bâti sur les briques décrites ici.

EN CLAIR

Ce papier n'a pas construit l'intelligence artificielle moderne. Il a retiré le goulot d'étranglement qui l'empêchait de grandir. Tout ce que le domaine a fait depuis consiste, pour l'essentiel, à verser du calcul et des données dans le moule qu'il a taillé.

Un dernier élément explique sa diffusion, et il est rarement mentionné : le papier est **court, clair et reproductible**. Douze pages, une figure d'architecture lisible, les hyperparamètres donnés, le code publié dans la foulée. Une idée juste qui reste inaccessible ne se répand pas ; celle-ci était à la portée de n'importe quelle équipe disposant de quelques cartes graphiques. C'est aussi pour cela qu'elle a gagné si vite.

Il est aujourd'hui l'un des articles scientifiques les plus cités de l'informatique. Le lire ne relève pas de la curiosité historique : la quasi-totalité des comportements, des limites et des coûts des systèmes actuels s'expliquent par des décisions prises dans ces douze pages.

Le problème : la séquentialité coûtait trop cher

Jusqu'en 2017, les modèles de traduction reposaient sur des réseaux **récurrents** — LSTM, GRU. Leur principe : lire un mot, mettre à jour un état interne, lire le mot suivant, mettre à jour, et ainsi de suite. L'état porte la mémoire de tout ce qui précède.

Cette conception a deux défauts, et le papier les nomme précisément.

Elle interdit la parallélisation à l'intérieur d'un exemple. Le calcul de l'état à la position t exige l'état à la position $t-1$. On ne peut donc pas traiter les mots simultanément : le temps de calcul croît avec la longueur de la phrase, et les milliers de cœurs d'un GPU restent inutilisés. C'est, dit le papier, « critique aux longueurs de séquence importantes ».

Elle allonge le chemin entre deux mots éloignés. Pour qu'un mot en position 1 influence un mot en position 50, l'information doit franchir 49 étapes de calcul successives. À chaque étape, elle se dilue. Apprendre une dépendance longue revient à faire passer un signal dans une longue chaîne de relais — plus la chaîne est longue, plus le signal s'affaiblit.

Des mécanismes d'attention existaient déjà, mais *en complément* de la récurrence : ils aidaient le décodeur à regarder la phrase source. La proposition du papier est plus radicale — supprimer la récurrence et ne garder que l'attention.

L'attention, sans les mathématiques d'abord

L'attention se décrit en trois rôles. Chaque mot de la phrase produit trois vecteurs :

- une **requête** (*query*) — ce que ce mot cherche ;
- une **clé** (*key*) — ce que ce mot annonce, ce à quoi il répond ;
- une **valeur** (*value*) — ce que ce mot apporte s'il est retenu.

Pour calculer la nouvelle représentation d'un mot, on compare sa requête à toutes les clés de la phrase. Chaque comparaison donne un score, les scores sont normalisés en poids, et la sortie est la moyenne des valeurs pondérée par ces poids. Un mot dont la clé répond bien à la requête pèse lourd ; les autres s'effacent.

EN CLAIR

Une bibliothèque. La *requête* est votre question au comptoir. Les *clés* sont les étiquettes au dos des livres. Les *valeurs* sont les contenus. Vous ne lisez pas toute la bibliothèque : vous comparez votre question aux étiquettes, vous retenez les trois ou quatre livres pertinents, et vous composez votre réponse à partir d'eux. L'attention fait cela pour chaque mot, contre tous les autres mots, en une seule opération.

La formule, et le détail qui compte

Le papier appelle son mécanisme *Scaled Dot-Product Attention*, l'attention par produit scalaire mis à l'échelle :

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

Les requêtes, clés et valeurs sont empilées en matrices, ce qui permet de tout calculer d'un coup. Le produit QK^T donne tous les scores. La fonction *softmax* les transforme en poids qui somment à 1. On multiplie enfin par V .

Reste le $\sqrt{d_k}$ au dénominateur, et c'est le détail qui a un nom dans le papier. Quand la dimension des clés grandit, les produits scalaires prennent des valeurs de plus en plus grandes en valeur absolue. Or la *softmax* mal se comporte sur de grandes valeurs : elle sature, met presque tout le poids sur un seul terme, et ses gradients deviennent extrêmement faibles. Diviser par la racine de la dimension ramène les scores à une échelle où la fonction reste sensible. Sans cette division, le modèle apprend mal.

POURQUOI PAS L'ATTENTION ADDITIVE

Le papier compare son choix à l'*attention additive*, qui calcule le score par un petit réseau plutôt que par un produit scalaire. Les deux ont une complexité théorique comparable, mais le produit scalaire s'écrit comme une multiplication de matrices — l'opération que le matériel exécute le plus vite. En pratique, il est « bien plus rapide et bien plus économe en mémoire ». Le choix est matériel, pas mathématique.

L'attention multi-tête : plusieurs regards, pas un seul

Une seule attention moyenne tout. Si un mot doit à la fois s'accorder avec son sujet grammatical et se rattacher à un thème lointain, une seule pondération ne peut pas faire les deux : elle produit un compromis.

La solution du papier est l'**attention multi-tête**. Plutôt qu'une attention en dimension complète, on projette les requêtes, clés et valeurs dans h sous-espaces plus petits, on calcule h attentions en parallèle, puis on concatène et on reprojette.

Dans le modèle de base : **$h = 8$ têtes**, une dimension de modèle $d_{\text{model}} = 512$, donc $d_k = d_v = 64$ par tête. Le papier le note : parce que chaque tête travaille en dimension réduite, le coût total reste voisin de celui d'une attention unique en pleine dimension.

EN CLAIR

Huit lecteurs relisent la même phrase, chacun avec sa consigne : l'un suit les accords, l'autre les pronoms, un troisième le fil du sujet. On met ensuite leurs annotations bout à bout. Une seule tête aurait produit un résumé mou de leurs huit lectures.

Trois emplois de l'attention dans une même architecture

Le Transformer est un encodeur-décodeur : l'encodeur lit la phrase source, le décodeur produit la traduction. L'attention y sert à trois endroits distincts, et les distinguer évite l'essentiel des confusions.

EMPLOI	CE QUE LE MODÈLE REGARDE
Auto-attention de l'encodeur	Chaque mot de la phrase source regarde tous les mots de la phrase source, lui-même compris. Rien n'est masqué : la phrase à traduire est connue en entier.
Auto-attention masquée du décodeur	Chaque mot déjà produit regarde les mots produits avant lui — et seulement eux . Les positions à venir sont masquées, mises à $-\infty$ avant la <i>softmax</i> . Sans ce masque, le modèle verrait la suite qu'il doit deviner.
Attention encodeur-décodeur	Les requêtes viennent du décodeur, les clés et valeurs de l'encodeur. C'est le pont : chaque mot en cours de production interroge toute la phrase source. C'est la forme d'attention qui existait déjà avant 2017.

Encodeur et décodeur comptent chacun **six couches identiques empilées**. Chaque couche d'encodeur contient une auto-attention puis un réseau position par position ; chaque couche de décodeur ajoute entre les deux l'attention encodeur-décodeur.

Ce qu'il y a autour de l'attention

L'attention seule ne suffit pas. Trois éléments l'entourent, moins spectaculaires et tout aussi nécessaires.

Un réseau position par position. Après l'attention, chaque position traverse le même petit réseau à deux couches, appliqué séparément et identiquement à chaque mot. Sa dimension intérieure vaut $d_{\text{ff}} =$

2048 pour un modèle en 512 — quatre fois plus large. L'attention mélange l'information entre positions ; ce réseau la transforme à l'intérieur de chaque position.

Les connexions résiduelles. Autour de chaque sous-couche, la sortie est ajoutée à l'entrée. Le gradient dispose ainsi d'un chemin direct vers les couches basses, ce qui rend l'empilement profond entraînable.

La normalisation de couche. Appliquée après chaque addition résiduelle, elle stabilise les échelles d'activation d'une couche à l'autre.

DÉTAIL QUI A BEAUCOUP COMPTÉ DEPUIS

Le papier place la normalisation *après* la sous-couche et l'addition résiduelle — ce qu'on appelle aujourd'hui *post-norm*. Les implémentations modernes la placent le plus souvent *avant* (*pre-norm*), disposition qui s'entraîne plus facilement en très grande profondeur. C'est l'un des rares points d'architecture où l'usage actuel s'écarte franchement du papier d'origine.

L'encodage positionnel : rendre l'ordre au modèle

En supprimant la récurrence, on supprime aussi la notion d'ordre. Pour l'attention pure, « le chat mange la souris » et « la souris mange le chat » sont le même sac de mots. Il faut réinjecter la position.

Le papier ajoute à chaque plongement de mot un **encodage positionnel** construit à partir de sinus et de cosinus de fréquences différentes : les dimensions paires reçoivent un sinus, les impaires un cosinus, et la longueur d'onde croît géométriquement le long des dimensions.

Deux raisons sont avancées. D'abord, pour tout décalage fixe k , l'encodage de la position $\text{pos} + k$ s'exprime comme une fonction linéaire de celui de pos — le modèle peut donc apprendre à raisonner sur des positions *relatives*. Ensuite, cette formulation permet d'espérer une extrapolation à des séquences plus longues que celles vues à l'entraînement.

Les auteurs ont aussi testé des encodages positionnels *appris* et rapportent des résultats presque identiques. Ils ont conservé la version sinusoïdale pour la seule raison d'extrapolation.

EN CLAIR

On tamponne chaque mot avec un motif ondulé propre à sa place dans la phrase. Le modèle ne lit pas « position 7 » comme un numéro : il perçoit une signature, et il apprend que deux signatures voisines correspondent à des mots voisins.

Pourquoi l'auto-attention, chiffres à l'appui

Le papier consacre une section entière à justifier le choix, et il le fait sur trois critères mesurables — c'est l'un des passages les plus utiles à retenir.

TYPE DE COUCHE	COMPLEXITÉ PAR COUCHE	OPÉRATIONS SÉQUENTIELLES	CHEMIN MAXIMAL
Auto-attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Récurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutionnel	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(\log_k n)$
Auto-attention restreinte (voisinage de taille r)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n / r)$

n est la longueur de la séquence, d la dimension des représentations, k la taille du noyau de convolution.

Deux lectures, opposées, sortent de ce tableau.

La bonne nouvelle tient dans les deux dernières colonnes. L'auto-attention demande un nombre *constant* d'opérations séquentielles — tout se parallélise — et le chemin entre deux positions quelconques est de longueur constante : n'importe quel mot atteint n'importe quel autre en une seule étape. C'est exactement ce que la récurrence ne savait pas faire.

La mauvaise est dans la première colonne : $O(n^2 \cdot d)$. Le coût croît avec le *carré* de la longueur. Le papier note que l'auto-attention est plus rapide que la récurrence tant que la longueur de séquence reste inférieure à la dimension des représentations — ce qui est « le cas le plus fréquent » pour les phrases traduites en 2017.

Ce coût quadratique est la contrainte structurelle du Transformer. Neuf ans de recherche sur les fenêtres de contexte longues — attentions creuses, linéaires, à fenêtre glissante, ré-encodage mémoriel — descendent en droite ligne de cette case du tableau. Le papier l'avait vue, et proposait déjà l'auto-attention restreinte à un voisinage comme parade.

Les auteurs avancent un troisième argument, plus discret : l'**interprétabilité**. Les poids d'attention sont observables, et ils rapportent que certaines têtes semblent adopter des comportements liés à la structure syntaxique de la phrase.

Comment il a été entraîné

Le protocole est décrit assez précisément pour être reproduit, ce qui n'allait pas de soi à l'époque.

ÉLÉMENT	VALEUR
Jeux de données	WMT 2014 anglais-allemand, 4,5 millions de paires de phrases ; anglais-français, 36 millions de paires
Vocabulaire	Encodage en sous-mots partagé source-cible
Matériel	8 GPU NVIDIA P100
Modèle de base	100 000 pas, environ 12 heures
Grand modèle	300 000 pas, 3 jours et demi
Optimiseur	Adam, $\beta_1 = 0,9 \cdot \beta_2 = 0,98 \cdot \varepsilon = 10^{-9}$
Pas d'apprentissage	Montée linéaire sur 4 000 pas, puis décroissance en l'inverse de la racine du numéro de pas
Régularisation	<i>Dropout</i> 0,1 sur le modèle de base ; lissage d'étiquettes $\varepsilon = 0,1$

Deux points méritent d'être soulignés. La **montée en température du pas d'apprentissage** — le fameux *warmup* — est devenue un standard de fait pour l'entraînement des Transformers, et son absence rend souvent l'entraînement instable. Et le **lissage d'étiquettes** : les auteurs notent qu'il *dégrade* la perplexité, parce qu'il apprend au modèle à être moins sûr de lui, mais qu'il améliore à la fois la précision et le score BLEU. On échange de la confiance contre de la justesse.

Les résultats

TÂCHE	SCORE BLEU	CE QUE ÇA VAUT
WMT 2014 anglais → allemand	28,4	Plus de 2 points au-dessus du meilleur résultat antérieur, ensembles compris
WMT 2014 anglais → français	41,8	Record pour un modèle unique, à une fraction du coût d'entraînement

Le second chiffre porte l'argument réel du papier. Battre un record est une chose ; le battre en trois jours et demi sur huit cartes, quand les modèles concurrents demandaient des semaines, en est une autre. C'est cette dernière propriété — *le même résultat, beaucoup moins cher* — qui a fait basculer le domaine en moins de deux ans.

Les auteurs testent enfin l'architecture sur une tâche entièrement différente, l'analyse syntaxique en constituants de l'anglais, avec peu d'ajustements. Le modèle y obtient de bons résultats, y compris en régime de données restreint. C'est un argument de généralité : le Transformer n'est pas un dispositif de traduction, c'est une architecture de séquences.

Les ablations, ou ce que les auteurs ont cassé exprès

Une table entière est consacrée aux variantes, et elle vaut mieux que bien des commentaires ultérieurs :

- **Trop peu ou trop de têtes dégrade.** Avec une seule tête, le modèle perd environ 0,9 BLEU par rapport au meilleur réglage ; avec beaucoup trop de têtes, la qualité redescend aussi. Il existe un optimum, il n'est pas au bord.
- **Réduire la dimension des clés dégrade.** Les auteurs en concluent que « déterminer la compatibilité n'est pas facile » et qu'une fonction de score plus élaborée qu'un produit scalaire pourrait être bénéfique. C'est une réserve du papier lui-même, rarement citée.
- **Plus grand est meilleur**, et le *dropout* aide beaucoup.
- **Encodages positionnels appris ou sinusoïdaux : résultats presque identiques.**

Ce qui a vieilli, ce qui tient

Neuf ans plus tard, la distinction est nette — et elle est ce qu'un lecteur d'aujourd'hui doit emporter.

ÉLÉMENT DU PAPIER	ÉTAT EN 2026
Attention par produit scalaire mise à l'échelle	Tient. C'est encore l'opération centrale de tous les grands modèles.
Attention multi-tête	Tient , avec des variantes d'efficacité qui partagent clés et valeurs entre plusieurs têtes pour alléger la mémoire à l'inférence.
Résiduels et normalisation	Tiennent , mais la normalisation est passée en amont (<i>pre-norm</i>) dans la quasi-totalité des implémentations.
Encodage positionnel sinusoïdal	Largement remplacé par des encodages de position relative, notamment rotatifs, qui se comportent mieux sur les contextes longs.
Structure encodeur-décodeur	Minoritaire. Les modèles de langage génératifs n'ont gardé que le décodeur. L'encodeur-décodeur reste employé pour les tâches de transduction stricte.
Coût quadratique en longueur	Toujours là. C'est la contrainte de fond derrière tout le travail sur les fenêtres de contexte.
Échelle : 8 GPU, 3,5 jours	Sans commune mesure. Les modèles de frontière actuels s'entraînent sur des dizaines de milliers d'accélérateurs pendant des mois.

EN CLAIR

Le mécanisme a tenu, l'emballage a changé. Si vous ne retenez qu'une chose du papier, retenez l'attention par produit scalaire et le tableau de complexité : le premier explique comment les modèles actuels lisent, le second explique pourquoi la longueur de contexte coûte si cher.

Ce que le papier ne dit pas

Trois précisions, parce que beaucoup lui est attribué à tort.

Il n'invente pas l'attention. Le mécanisme existait depuis 2014-2015 dans les modèles de traduction récurrents. Ce que fait le papier, c'est le rendre *suffisant* : il retire tout le reste.

Il ne parle ni de lois d'échelle, ni de modèles de langage généralistes, ni d'émergence. Son horizon est la traduction automatique et, accessoirement, l'analyse syntaxique. L'idée qu'on puisse faire grandir cette architecture jusqu'à obtenir des systèmes généralistes viendra plus tard, et d'autres équipes.

Il ne prétend pas que l'attention est interprétable. Il observe que certaines têtes *semblent* adopter des comportements liés à la syntaxe. La question de savoir si les poids d'attention expliquent vraiment une décision du modèle a fait l'objet d'une littérature entière depuis, et elle n'est pas tranchée.

Comment le lire aujourd'hui

Le papier fait douze pages et reste accessible — c'est un texte d'ingénierie, pas de mathématiques avancées. Trois parcours selon ce que vous cherchez :

VOUS VOULEZ	LISEZ
Comprendre le mécanisme	La section sur l'attention, en particulier le produit scalaire mis à l'échelle et l'attention multi-tête. C'est le cœur, tout le reste en découle.
Décider d'une architecture	Le tableau de complexité. Trois lignes, trois colonnes, et l'essentiel des arbitrages de conception des dix dernières années.
Reproduire ou entraîner	La section d'entraînement — montée du pas d'apprentissage, lissage d'étiquettes, <i>dropout</i> — puis la table des variantes, qui dit ce qui casse quand on y touche.

Une remarque pour finir. Le titre du papier est une plaisanterie de laboratoire qui a très mal vieilli, dans un sens précis : l'attention n'est pas tout ce dont on a besoin. Il faut aussi les résiduels, la normalisation, l'encodage positionnel, le réseau position par position, la montée du pas d'apprentissage et le lissage d'étiquettes. Retirez n'importe lequel de ces éléments et l'entraînement se dégrade ou échoue. Le papier lui-même le montre, table de variantes à l'appui. Ce que l'attention a remplacé, c'est la *récurrence* — pas le reste de l'ingénierie.

Source primaire — A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, « Attention Is All You Need », NIPS 2017. arXiv:1706.03762 — <https://arxiv.org/abs/1706.03762>

Les scores BLEU (28,4 et 41,8), la durée d'entraînement (3,5 jours sur huit GPU), les hyperparamètres et le tableau de complexité sont cités depuis le papier. Les mentions de l'état de l'art 2026 relèvent de la synthèse et sont signalées comme telles dans le texte.