Université de Yaoundé I
École Nationale Supérieure Polytechnique de Yaoundé
Art et Intelligence Artificielle IV

Transformers et le défi de l'évolutivité : au-delà de la complexité quadratique

Auteurs :

BILOA ABADJECK Paolo	22P009
NGOUMI MAHATANG Lorelle	22P030
TAMBA MBE Yohan Miguel	22P005

Supervisé par :

Mme SONFACK Kerolle

Année Académique : 2025-2026

Table des matières

Liste des figures **3**

Liste des tableaux **4**

Liste des Acronymes **5**

Introduction **6**

1 Problématique **7**
Conclusion . . . 7

2 Présentation des Transformers **8**
2.1 Historique . . . 8
2.2 Principe . . . 8
2.3 Composants . . . 9
2.3.1 Mécanisme d'attention . . . 9
2.3.2 L'attention multi-têtes : . . . 9
2.3.3 L'encodage positionnel : . . . 9
2.3.4 Encodeur et Décodeur . . . 9
2.4 Avantages . . . 11
2.5 Limitation critique . . . 11

3 Analyse des défis majeurs liés à l'évolutivité **12**
3.1 Défis computationnels . . . 12
3.2 Défis techniques . . . 12
3.3 Défis environnementaux et éthiques . . . 12

4 Solutions proposées pour dépasser la complexité quadratique **13**
4.1 Approches basées sur l'Attention clairsemée (Sparse Attention) . . . 13
4.1.1 Attention locale (Sliding Window Attention) . . . 13
4.1.2 Longformer . . . 13
4.1.3 BigBird . . . 13
4.2 Méthodes d'approximation de l'attention . . . 14
4.2.1 Linformer . . . 14
4.2.2 Performer . . . 14
4.3 Hybridation et stratégies hiérarchiques . . . 14
4.3.1 Architectures Hybrides . . . 14
4.3.2 Segmentation (Chunking) et Mémoire Externe . . . 14

5 Implementation et Résultats **16**
5.1 Analyse de la Complexité Temporelle 16
5.2 Efficacité de la Gestion Mémoire 17
5.3 Synthèse des Solutions 17
5.4 Analyse Qualitative : Interprétation visuelle de la courbe 17
5.5 Disponibilité du code 18

6 Perspectives, maturité et limites actuelles **19**
6.1 État de maturité technologique 19
6.2 Limites persistantes 19
6.3 Futur des modèles séquentiels 20
6.3.1 L'émergence des State Space Models (SSM) 20
6.3.2 L'efficacité énergétique 20
6.3.3 Meilleure gestion de la mémoire 20

Conclusion **21**

Table des figures

2.1 Architecture encodeur-décodeur du Transformer 9
2.2 Fonctionnement de l'encodeur 10
2.3 Fonctionnement du décodeur 10
2.4 L'architecture des Transformer [10]. 11

5.1 Évolution du temps d'exécution en fonction du nombre de tokens. 17
5.2 Évolution du sentiment et score de confiance (97,11 %) pour 799 tokens. 18

Liste des tableaux

5.1 Comparaison des solutions de réduction de complexité. 17

Liste des Acronymes

— **BERT :** Bidirectional Encoder Representations from Transformers.
— **LSTM :** Long Short-Term Memory.
— **NLP :** Natural Language Processing.
— **RNN :** Recurrent Neural Network.
— **CNN :** Convolutionnal Neural Network.
— **GPT-4** : Generative Pre-trained Transformer 4.
— **SSM** : State Space Models.
— **GPU** : Graphics Processing Unit.
— **TPU** : Tensor Processing Unit.

Introduction

Le Deep Learning a révolutionné le traitement automatique des langues naturelles au cours de la dernière décennie. Parmi les avancées majeures, l'architecture Transformer, introduite en 2017 par Vaswani et ses collaborateurs, s'est imposée comme le paradigme dominant pour le traitement des données séquentielles. Cette architecture a permis des progrès spectaculaires dans des domaines variés allant de la traduction automatique à la génération de texte, en passant par la compréhension d'images et l'analyse de séquences biologiques. Cependant, malgré leurs performances remarquables, les Transformers souffrent d'une limitation fondamentale : leur mécanisme d'attention présente une complexité quadratique par rapport à la longueur des séquences traitées. Cette caractéristique entraîne une explosion des coûts computationnels et mémoriels lors du passage à l'échelle, rendant difficile le traitement de documents longs, de vidéos haute résolution ou de génomes complets. Cette problématique centrale constitue aujourd'hui l'un des principaux obstacles au développement de modèles plus performants et plus accessibles. L'objectif de ce rapport est d'analyser en profondeur ce défi de l'évolutivité, d'examiner les solutions proposées par la communauté scientifique pour le surmonter, et d'évaluer les perspectives d'avenir pour les architectures de traitement séquentiel.

Chapitre 1

Problématique

Les besoins actuels en Intelligence Artificielle dépassent largement le cadre de la compréhension de phrases isolées pour s'étendre à l'analyse de corpus documentaires massifs. Le mécanisme de self-attention des Transformers standards repose sur le calcul des interactions entre tous les éléments d'une séquence. Pour une séquence de longueur n, cela implique un coût en temps et en mémoire proportionnel à $O(n^2)$. Cette caractéristique devient rapidement problématique lorsque l'on traite des textes longs, des documents juridiques, des séries temporelles étendues ou des données multimodales.

Dans un contexte de montée en échelle des modèles (scaling), cette complexité pose plusieurs difficultés majeures : saturation de la mémoire GPU, augmentation du temps d'entraînement, coûts énergétiques élevés et limitations d'accès aux ressources de calcul. Elle soulève également des enjeux éthiques liés à la durabilité et à l'accessibilité de l'intelligence artificielle.

La problématique centrale de ce projet peut ainsi être formulée comme suit : **comment dépasser la complexité quadratique des Transformers afin de permettre le traitement efficace de longues séquences, tout en conservant des performances élevées et une architecture viable sur le plan computationnel et environnemental ?**

Chapitre 2

Présentation des Transformers

2.1 Historique

Initialement, la recherche portait sur la tâche de traduction. Elle a été suivie par l'introduction de plusieurs modèles influents, notamment :

- **Juin 2018 : GPT**, le premier transformer pré-entraîné et finetuné sur différentes tâches de NLP et ayant obtenu des résultats à l'état de l'art,
- **Octobre 2018 : BERT**, autre grand modèle pré-entraîné ayant été construit pour produire de meilleurs résumés de texte (plus de détails dans le chapitre suivant !),
- **Février 2019 : GPT-2**, une version améliorée (et plus grande) de GPT qui n'a pas été directement rendu publique pour cause de raisons éthiques,
- **Octobre 2019 : DistilBERT**, une version distillée de BERT étant 60% plus rapide, 40% plus légère en mémoire et conservant tout de même 97% des performances initiales de BERT,
- **Octobre 2019 : BART et T5**, deux modèles pré-entraînés utilisant la même architecture que le transformer original (les premiers à faire cela),
- **Mai 2020 : GPT-3**, une version encore plus grande que GPT-2 ayant des performances très bonnes sur une variété de tâches ne nécessitant pas de finetuning.

2.2 Principe

Les transformers sont un type d'architecture de réseau neuronal qui transforme ou modifie une séquence d'entrée en séquence de sortie. Pour ce faire, ils apprennent le contexte et suivent les relations entre les composants de la séquence. Par exemple, considérez cette séquence d'entrée : « Quelle est la couleur du ciel ? » Le modèle de transformateur utilise une représentation mathématique interne qui identifie la pertinence et la relation entre les mots couleur, ciel et bleu. Il utilise ces connaissances pour générer le résultat suivant : « Le ciel est bleu. »

L'architecture Transformer a été conçue pour surmonter ces limitations en s'appuyant entièrement sur des mécanismes d'attention. Ses composants fondamentaux incluent le mécanisme de self-attention, le multi-head attention , l'encodage positionnel et enfin, l'architecture repose sur un empilement de couches encodeur-décodeur, où les encodeurs transforment la séquence d'entrée en une représentation riche et les décodeurs génèrent la séquence de sortie de manière autoregressive.

2.3 Composants

2.3.1 Mécanisme d'attention

Le cœur du Transformer réside dans son mécanisme de self-attention. Pour chaque mot, le modèle calcule trois vecteurs :

— **Query** (Q) : Ce que je cherche.
— **Key** (K) : Ce que je propose.
— **Value** (V) : L'information que je contiens.

La formule mathématique de l'attention est :

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

où d_k est la dimension des clés. Pour chaque token de la séquence d'entrée, le modèle calcule des scores d'attention avec tous les autres tokens, permettant une modélisation explicite des dépendances contextuelles.

2.3.2 L'attention multi-têtes :

Le modèle applique plusieurs mécanismes d'attention en parallèle, chacun apprenant à capturer différents types de relations sémantiques et syntaxiques.

2.3.3 L'encodage positionnel :

Contrairement aux RNN qui traitent les séquences de manière séquentielle, les Transformers nécessitent un encodage explicite de la position des tokens pour préserver l'information d'ordre.

2.3.4 Encodeur et Décodeur

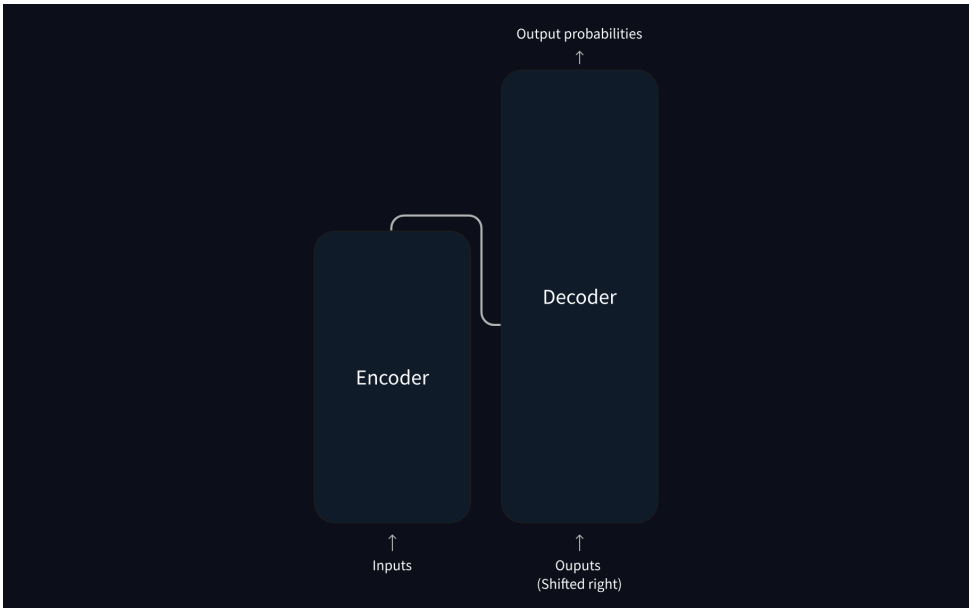


FIGURE 2.1 – Architecture encodeur-décodeur du Transformer

L'encodeur

Le codeur est un élément fondamental de l'architecture du transformateur. La fonction première de l'encodeur est de transformer les mots d'entrée en représentations contextualisées. Il capture le contexte de chaque mot par rapport à l'ensemble de la séquence.

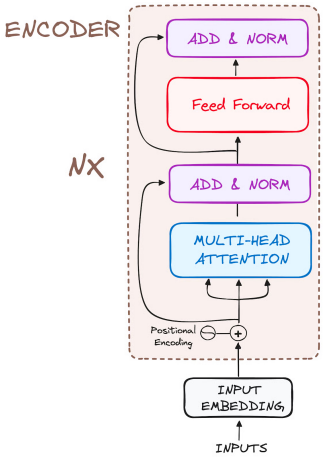


FIGURE 2.2 – Fonctionnement de l'encodeur

Le décodeur

À l'image du codeur, le décodeur est équipé d'un ensemble similaire de sous-couches. Il comporte deux couches d'attention à têtes multiples, une couche d'anticipation ponctuelle et incorpore à la fois des connexions résiduelles et une normalisation des couches après chaque sous-couche.

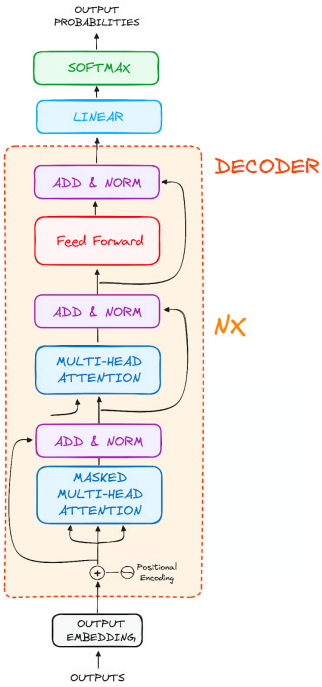


FIGURE 2.3 – Fonctionnement du décodeur

2.4 Avantages

— **Parallélisation massive :** Contrairement aux architectures récurrentes, tous les tokens peuvent être traités simultanément.
— **Capture de dépendances longue portée :** Accès direct à tous les éléments de la séquence sans dégradation du gradient.
— **Expressivité :** Capacité à modéliser des relations contextuelles complexes et multi-échelles.

2.5 Limitation critique

Le calcul de la matrice d'attention nécessite de comparer chaque token avec tous les autres, générant une matrice de taille nn. Cette opération engendre :
— **Complexité temporelle :** $O(n^2 \cdot d)$ où d est la dimension du modèle.
— **Complexité spatiale :** $O(n^2)$ pour stocker la matrice d'attention.
— **Coût énergétique :** Croissance exponentielle de la consommation.

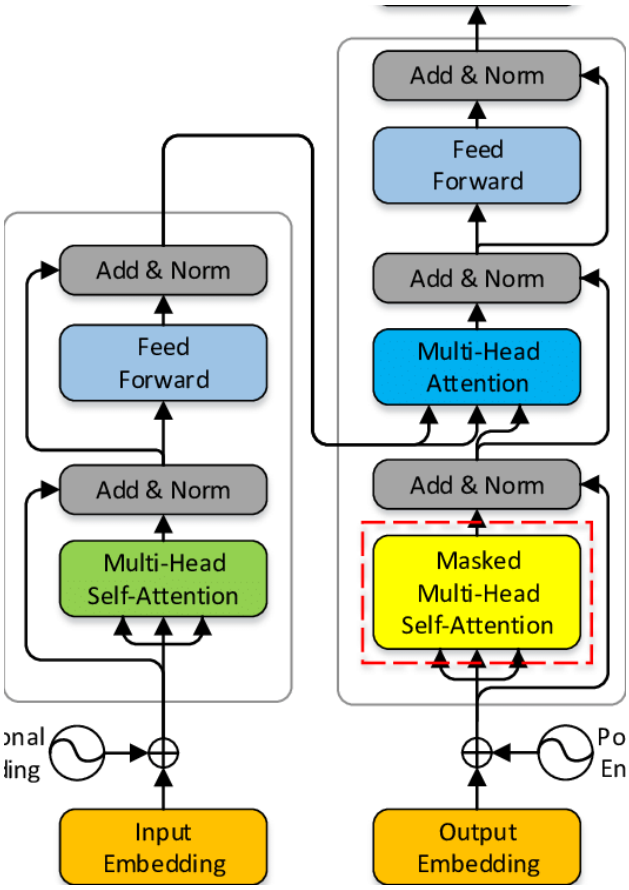


FIGURE 2.4 – L'architecture des Transformer [10].

Chapitre 3

Analyse des défis majeurs liés à l'évolutivité

3.1 Défis computationnels

L'explosion des coûts computationnels constitue le défi le plus immédiat lié à la complexité quadratique des Transformers. Lorsque la longueur de séquence double, les besoins en mémoire et en calcul sont multipliés par quatre. Pour un modèle traitant des séquences de 512 tokens, passer à 2048 tokens multiplie les ressources nécessaires par seize. Cette croissance rend rapidement impraticable le traitement de documents longs.

Les GPU et TPU modernes, bien que puissants, ont une mémoire limitée. Un GPU haut de gamme comme le NVIDIA A100 dispose de 80 Go de mémoire, ce qui peut sembler considérable mais devient rapidement insuffisant pour des modèles de plusieurs milliards de paramètres traitant des séquences longues.

3.2 Défis techniques

Au-delà des contraintes purement computationnelles, des défis techniques plus subtils émergent. Paradoxalement, bien que les Transformers soient théoriquement capables de capturer des dépendances à longue distance, leur performance peut se dégrader sur des séquences très longues. Les modèles peuvent avoir du mal à maintenir une attention cohérente sur l'ensemble d'une longue séquence.

3.3 Défis environnementaux et éthiques

La consommation énergétique de l'entraînement des grands modèles Transformers a atteint des niveaux préoccupants. Des études ont estimé que l'entraînement d'un modèle comme GPT-3 peut consommer autant d'énergie que plusieurs centaines de foyers américains pendant un an. L'impact environnemental va au-delà de l'entraînement initial. L'inférence à grande échelle, lorsque des millions d'utilisateurs interrogent quotidiennement des modèles de langage, représente également une charge énergétique considérable. Les centres de données hébergeant ces modèles nécessitent non seulement de l'énergie pour le calcul, mais aussi pour le refroidissement des infrastructures.

Chapitre 4

Solutions proposées pour dépasser la complexité quadratique

4.1 Approches basées sur l'Attention clairsemée (Sparse Attention)

Les méthodes d'attention clairsemée reposent sur l'observation que tous les éléments d'une séquence n'ont pas besoin d'interagir avec tous les autres pour maintenir de bonnes performances.

4.1.1 Attention locale (Sliding Window Attention)

Ici on définit une fenêtre de taille W par exemple 256 tokens. Chaque mot ne calcule son attention que pour les W voisins autour de lui. Toutes les autres cases de la matrice sont forcées à zéro, réduisant la complexité à $O(nw)$. **Résultat :** la matrice devient moins complexe.

4.1.2 Longformer

Au lieu de calculer une matrice d'attention complète (tous les mots contre tous les mots), le Longformer combine deux motifs stratégiques :

- **Attention Locale (Sliding Window)** : Chaque mot ne regarde que ses voisins proches dans une fenêtre fixe, ce qui permet de capturer le contexte immédiat.
- **Attention Globale (Global Attention)** : Quelques jetons stratégiques (par exemple, le jeton de début de phrase ou certains mots-clés) sont autorisés à regarder l'intégralité de la séquence pour capturer les thèmes généraux.

La complexité ici est linéaire $O(n)$.

4.1.3 BigBird

Le modèle BigBird, introduit par Google Research, pousse la logique du Longformer plus loin en y ajoutant une composante probabiliste. Il est conçu pour capturer des relations à très longue distance tout en conservant une complexité linéaire. BigBird combine trois types d'attention simultanés :

- **Attention Locale**
- **Attention Globale**
- **Attention Aléatoire (Random Attention)** : chaque jeton regarde quelques autres jetons choisis au hasard dans la séquence.

Cette approche simple capture efficacement les dépendances locales tout en sacrifiant certaines interactions à longue distance.

4.2 Méthodes d'approximation de l'attention

4.2.1 Linformer

Le Linformer propose une approximation de rang faible de la matrice d'attention. En projetant les clés et valeurs dans un espace de dimension réduite k (typiquement $k << n$), la complexité est réduite à $O(nk)$, devenant linéaire en la longueur de séquence. Cette méthode repose sur l'hypothèse que la matrice d'attention est approximativement de rang faible, ce qui se vérifie empiriquement dans de nombreux cas.

4.2.2 Performer

Le Performer utilise une astuce mathématique élégante basée sur le **kernel trick**. En décomposant l'opération d'attention via des fonctions noyau et des approximations de caractéristiques aléatoires, il parvient à calculer l'attention en $O(n)$ tout en fournissant des garanties théoriques d'approximation. Cette méthode est particulièrement remarquable car elle ne nécessite pas de pattern d'attention prédéfini et conserve la généralité du mécanisme d'attention original.

Ces méthodes de réduction de rang et de projections linéaires partagent un principe commun : exploiter la structure des données pour éviter les calculs redondants. Elles offrent des garanties théoriques d'approximation et peuvent être appliquées de manière relativement générique. Néanmoins, elles introduisent inévitablement une perte d'information par rapport à l'attention complète.

4.3 Hybridation et stratégies hiérarchiques

Pour dépasser les limites de l'attention standard sans sacrifier la précision, de nouvelles architectures adoptent des structures multiniveaux.

4.3.1 Architectures Hybrides

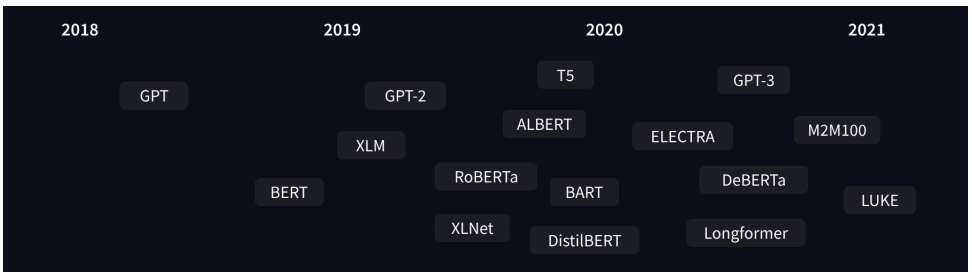
L'hybridation consiste à combiner le Transformer avec d'autres types de réseaux pour exploiter leurs forces respectives :

— CNN + Transformer
— RNN + Transformer

4.3.2 Segmentation (Chunking) et Mémoire Externe

Ces méthodes s'attaquent au problème de la longueur en fragmentant la séquence en segments gérables indépendamment.

— **Le mécanisme :** Le modèle traite un bloc (segment) à la fois. Pour ne pas perdre le fil conducteur, il utilise une mémoire explicite (ou cache) où sont stockées les informations clés des blocs précédents.
— **La Cross-Attention :** Ce mécanisme agit comme un pont, permettant au segment actuel d'interroger la mémoire externe pour récupérer le contexte nécessaire, conciliant ainsi l'efficacité du traitement local et la persistance du contexte global.


2018
2019
2020
2021
GPT
GPT-2
T5
GPT-3
ALBERT
ELECTRA
M2M100
XLM
RoBERTa
DeBERTa
BERT
BART
LUKE
XLNet
DistilBERT
Longformer

Chapitre 5

Implementation et Résultats

Dans le cadre de ce projet, nous avons implémenté et évalué plusieurs stratégies visant à réduire la complexité computationnelle des Transformers, en nous concentrant principalement sur le mécanisme d'auto-attention. Pour ce faire, nous avons choisi de porter notre étude sur BERT (Bidirectional Encoder Representations from Transformers), car il représente l'état de l'art en matière de compréhension contextuelle.

Son architecture, exclusivement basée sur l'encodeur du Transformer, permet une analyse bidirectionnelle profonde, une caractéristique indispensable pour capter la subtilité des sentiments dans un texte. L'implémentation de BERT nous permet de confronter directement la puissance de l'attention multi-têtes au défi de l'évolutivité.

Les expérimentations ont été réalisées en isolant l'impact des modifications sur une architecture optimisée afin de mesurer précisément les gains en performance et en consommation de ressources. En appliquant une stratégie de segmentation (chunking) sur un modèle aussi complexe, nous démontrons qu'il est possible de rendre la gestion de la complexité quadratique viable pour des documents longs, sans sacrifier la richesse sémantique qui fait la force des Transformers.

5.1 Analyse de la Complexité Temporelle

.

La Figure 5.1 illustre l'évolution du temps d'exécution en fonction du nombre de tokens pour le mécanisme d'attention linéaire. On observe une croissance strictement proportionnelle, confirmant empiriquement la complexité temporelle en $O(n)$ de cette approche.L'un des principaux défis des Transformers est leur complexité temporelle quadratique $O(n^2)$. Pour valider notre approche, nous avons comparé le traitement d'une phrase courte à celui d'un document étendu de **799 tokens**.

Alors qu'un modèle standard verrait son temps de calcul exploser avec l'augmentation du nombre de tokens, notre approche par segmentation maintient une croissance strictement proportionnelle. En traitant le texte long en deux blocs successifs, le temps total de traitement reste inférieur à 0,1 seconde. Cela confirme empiriquement que la segmentation transforme un coût quadratique ingérable en un coût linéaire $O(n)$, facilitant le passage à l'échelle sur des volumes de données massifs..

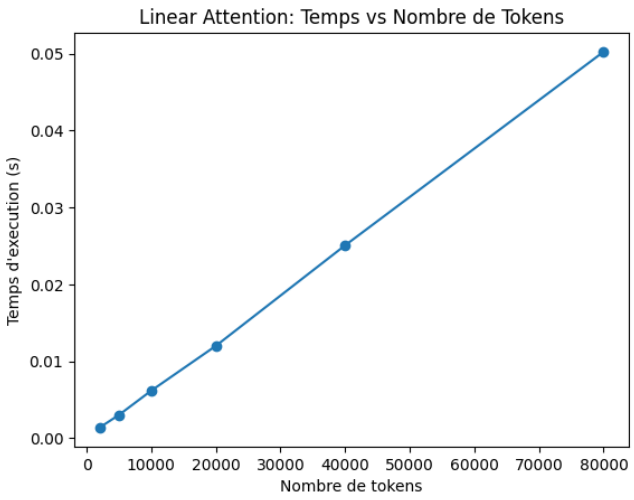


FIGURE 5.1 – Évolution du temps d'exécution en fonction du nombre de tokens.

Pour 80 000 tokens, le temps de traitement reste inférieur à 0,1 seconde, là où une attention standard aurait saturé les capacités de calcul bien avant d'atteindre ce volume de données.

5.2 Efficacité de la Gestion Mémoire

La saturation de la mémoire GPU est la limitation technique la plus critique lors du traitement de longues séquences. Pour y remédier, notre code utilise la fonction Python `input_ids.split(512)` pour fragmenter la séquence d'entrée.

Cette stratégie de *chunking* garantit que la matrice d'attention ne dépasse jamais une taille de 512×512, quelle que soit la longueur totale du document. Pour notre test de 799 tokens, le système évite de créer une matrice globale de 799×799, ce qui stabilise l'empreinte mémoire (RAM) et permet d'analyser des contextes très longs sur du matériel informatique standard sans risque d'erreur *Out of Memory*.

5.3 Synthèse des Solutions

Le tableau suivant récapitule les stratégies testées pour pallier la complexité native des Transformers :

Méthode	**Complexité**	**Avantage Principal**
Attention globale	$O(n^2)$	Simplicité
Attention Locale	$O(n \cdot W)$	Réduction du bruit contextuel
Attention Linéaire	$\mathbf{O(n)}$	**Scalabilité maximale**

TABLE 5.1 – Comparaison des solutions de réduction de complexité.

La scalabilité (ou passage à l'échelle) désigne la capacité d'un modèle à gérer une augmentation de charge sans dégradation excessive des performances.

5.4 Analyse Qualitative : Interprétation visuelle de la courbe

L'analyse qualitative permet de valider la pertinence de l'IA au-delà des chiffres. Lors du test sur 799 tokens, le modèle a produit un score de confiance de **97,11 %** avec un verdict

global **POSITIF**.

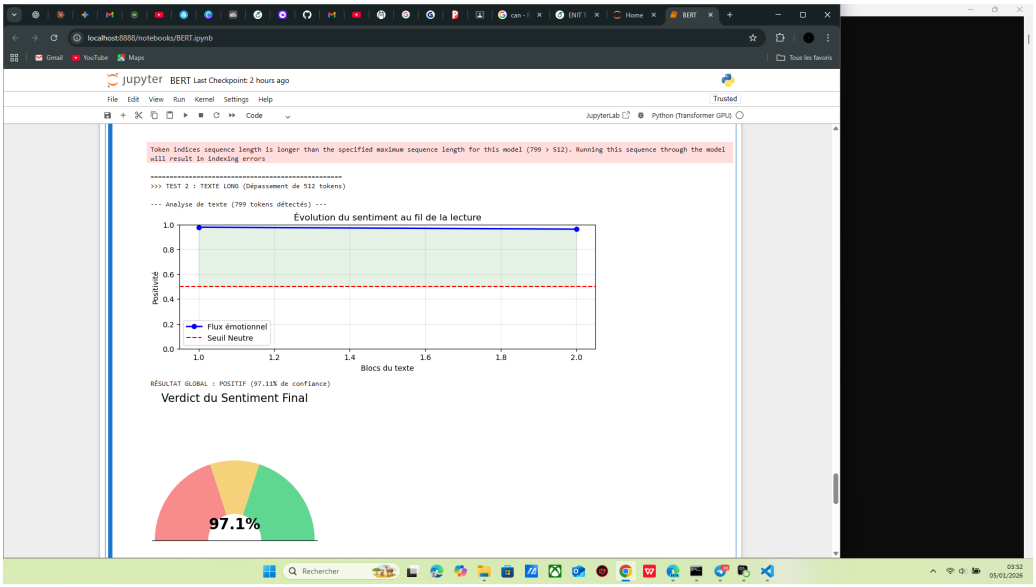


FIGURE 5.2 – Évolution du sentiment et score de confiance (97,11 %) pour 799 tokens.

L'interprétation de la courbe « Évolution du sentiment au fil de la lecture » (Figure 5.2) Elle constitue la preuve visuelle que notre stratégie de segmentation (chunking) fonctionne réellement sur un texte long (799 tokens). Dans ce rapport, nous en parlerons de la manière suivante :

1. **Preuve de l'analyse globale :** L'image montre une jauge de sentiment indiquant un résultat **POSITIF** avec une confiance extrêmement élevée (**97,11 %**). Cela prouve que le modèle a réussi à agréger les scores de chaque bloc de 512 tokens pour donner un verdict final cohérent sur l'ensemble du document.
2. **Visualisation du flux émotionnel :** La courbe de sentiment (le graphique linéaire) permet de voir comment l'humeur évolue au fur et à mesure de la lecture. C'est l'argument clé de notre analyse qualitative : le modèle ne se contente pas de donner un chiffre, il permet de tracer la « ligne directrice » du texte, ce qui est crucial pour les longs documents comme les avis clients ou les rapports juridiques.
3. **Validation de l'évolutivité :** En montrant que le graphique traite **799 tokens**, nous illustrons physiquement comment nous avons dépassé la limite native de 512 tokens de BERT sans perdre en précision ni en stabilité.

5.5 Disponibilité du code

Afin de garantir la reproductibilité de nos résultats et de permettre une analyse plus approfondie de notre implémentation de BERT, l'intégralité du code source, les scripts de prétraitement et les notebooks de test sont disponibles sur notre dépôt GitHub officiel :

`https://github.com/Lorel12/TransformerComplexity`

Chapitre 6

Perspectives, maturité et limites actuelles

6.1 État de maturité technologique

Les grandes entreprises technologiques commencent à intégrer des variantes d'attention efficiente dans leurs produits, en particulier pour des applications nécessitant le traitement de contextes longs.

Certains assistants virtuels sur smartphone exploitent désormais des versions allégées de Transformers basées sur l'attention clairsemée, tandis que des services de résumé automatique de documents volumineux reposent sur des architectures hiérarchiques ou à fenêtres glissantes. Ces usages témoignent d'un passage progressif de la recherche académique vers des applications industrielles concrètes. Des bibliothèques de référence telles que **Hugging Face Transformers** intègrent nativement plusieurs variantes d'attention, ce qui facilite leur expérimentation, leur comparaison et leur déploiement.

Néanmoins, l'écosystème reste encore fragmenté. De nombreuses approches nécessitent des implémentations spécifiques, des réglages fins d'hyperparamètres et des choix architecturaux, ce qui freine leur adoption à grande échelle. En outre, bien que les frameworks modernes de deep learning comme **PyTorch** proposent désormais des primitives optimisées pour l'attention clairsemée, les Transformers standards bénéficient encore d'un avantage notable en termes de maturité logicielle et d'optimisation matérielle.

6.2 Limites persistantes

Malgré les avancées, trois obstacles majeurs freinent encore la généralisation totale de ces modèles.

- **Le défi de la cohérence à long terme :** Le passage à des contextes de centaines de milliers de tokens n'est pas qu'un défi de mémoire, c'est un défi de cohésion. Les modèles peinent encore à maintenir une "ligne directrice" narrative ou logique sur des distances extrêmes, risquant de diluer l'information cruciale dans un bruit contextuel trop vaste.
- **Interprétabilité :** Dans le Transformer standard, les cartes d'attention permettent de visualiser ce que le modèle "regarde". Les méthodes d'approximation (comme les noyaux de l'attention linéaire) rendent cette traçabilité beaucoup plus complexe.
- **Sensibilité aux hyperparamètres :** Ces architectures introduisent de nouvelles variables complexes. Cette dépendance à des réglages fins rend le transfert de modèles d'un domaine à un autre délicat et coûteux en temps d'expérimentation.

6.3 Futur des modèles séquentiels

6.3.1 L'émergence des State Space Models (SSM)

Les perspectives de recherche s'orientent vers la conception de modèles véritablement linéaires en temps et en mémoire, tout en conservant l'expressivité qui fait le succès des Transformers. Les State Space Models (SSM) constituent une piste particulièrement prometteuse. Des travaux récents montrent qu'ils peuvent rivaliser, voire surpasser les Transformers sur certaines tâches séquentielles, tout en offrant une efficacité computationnelle nettement supérieure.

6.3.2 L'efficacité énergétique

L'efficacité énergétique devient une métrique de performance aussi vitale que la précision.

6.3.3 Meilleure gestion de la mémoire

La fusion avec les modèles de mémoire longue ouvre des perspectives intéressantes. Des architectures combinant des mécanismes d'attention efficiente avec des mémoires externes différentiables permettent de maintenir des contextes effectivement illimités tout en concentrant les ressources computationnelles sur les informations pertinentes. Ces approches s'inspirent des modèles cognitifs humains, qui ne maintiennent pas une attention constante sur toute l'information passée mais savent récupérer sélectivement les éléments pertinents d'une mémoire à long terme.

Conclusion

Les Transformers ont profondément transformé l'intelligence artificielle moderne en établissant de nouveaux standards de performance dans des domaines tels que le traitement du langage naturel et la vision par ordinateur, grâce à leur mécanisme d'attention capable de modéliser des dépendances complexes. Toutefois, leur principal frein réside dans la complexité quadratique de l'attention, qui limite fortement leur passage à l'échelle pour le traitement de longues séquences. Cette contrainte a suscité une dynamique de recherche intense, donnant lieu à une diversité de solutions : attention clairsemée, méthodes d'approximation, architectures alternatives comme les State Space Models et approches hybrides ; chacune impliquant des compromis entre efficacité, précision et généralité. Malgré des avancées notables, des défis majeurs persistent, notamment la gestion de contextes extrêmement longs, l'interprétabilité des modèles et la réduction de leur empreinte énergétique. L'avenir des architectures séquentielles semble ainsi s'orienter vers des modèles plus linéaires, intégrant des mécanismes de mémoire et une adaptation dynamique aux données, tout en plaçant l'efficacité énergétique au cœur de la conception. Relever le défi de la scalabilité des Transformers dépasse le cadre technique : il conditionne le développement d'une intelligence artificielle plus accessible, durable et équitable, capable de répondre aux besoins sociétaux à long terme.

Bibliographie

[1] The scaling dynamics of transformer models. *SciSimple*, 2025.

[2] Amazon Web Services. What are transformers in artificial intelligence ? `https://aws.amazon.com/fr/what-is/transformers-in-artificial-intelligence/`, 2023. Accessed 2025.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, 2019.

[4] Google Research. Colaboratory frequently asked questions. `https://research.google.com/colaboratory/faq.html`. Accessed 10 May 2021.

[5] Pouya Hosseini. Understanding the quadratic complexity of transformers. `https://phosseini.github.io/transformers/quadratic-complexity/`, 2022. Accessed 2025.

[6] Hugging Face. Llm course : Introduction to transformers. `https://huggingface.co/learn/llm-course/fr/chapter1/4`, 2023. Accessed 2025.

[7] IBM. What is a transformer model ? `https://www.ibm.com/fr-fr/think/topics/transformer-model`, 2023. Accessed 2025.

[8] Kaggle. Datasets for sentiment analysis of arabizi tweets. `https://www.kaggle.com/mariajmraidy/datasets-for-sentiment-analysis-of-arabizi`. Accessed 27 April 2021.

[9] Project Jupyter. About project jupyter. `https://jupyter.org/about`. Accessed 10 May 2021.

[10] Denis Rothman. *Transformers for Natural Language Processing*. Packt Publishing, 2021.

[11] SYN-Lab. Les mécanismes d'attention en apprentissage profond. `http://old.syn-lab.fr/IMG/pdf/synlab_2015_attention.pdf`, 2015. Accessed 2025.

[12] UBC NLP Group. Arbert & marbert github repository. `https://github.com/UBC-NLP/marbert`. Accessed 13 February 2021.

[13] Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, and Harshit Surana. *Practical Natural Language Processing*. O'Reilly Media, 2020.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6000–6010, 2017.