Synthèse de texte automatique avec les transformateurs

Par :

KHALID Hamza, OUAHBI Ismail, CHAFIKI Mohamed El Amine

Encadré par :

Pr. BOUZAACHANE Khadija

Abstract :

Dans cet article, nous proposons une nouvelle méthode de synthèse de texte à l'aide de modèles de langage basés sur des transformateurs. En tirant parti de la capacité de ces modèles à capturer les dépendances et le contexte à long terme, nous sommes en mesure de générer des résumés plus cohérents et précis par rapport aux méthodes traditionnelles. Nous évaluons notre méthode sur plusieurs ensembles de données de référence et montrons des améliorations significatives de la qualité du résumé. De plus, nous démontrons l'efficacité de notre approche sur des articles de presse du monde réel, démontrant son caractère pratique pour une variété d'applications.

Problématique :

La synthèse de texte, également connue sous le nom de résumé de texte, est un processus consistant à condenser un texte en un résumé plus court qui capture l'essentiel de l'information contenue dans le texte original. Cela peut être utile dans de nombreuses situations, telles que la préparation de rapports, la recherche de documents pertinents pour un projet ou la préparation de présentations.

Il existe de nombreux défis liés à la synthèse de texte. Tout d'abord, il peut être difficile de déterminer quelles informations sont les plus importantes et doivent être incluses dans le résumé. Deuxièmement, il peut être difficile de condenser un texte en un résumé plus court sans perdre l'essentiel de l'information. Enfin, il peut être difficile de rédiger le résumé de manière à ce qu'il soit clair et cohérent, tout en étant concis.

Il existe plusieurs approches pour résumer un texte, y compris l'utilisation de techniques de synthèse automatique, comme l'analyse de mots-clés ou la détection de phrases importantes, ou en utilisant des algorithmes de machine learning pour prédire les informations les plus importantes dans un texte. Cependant, la synthèse de texte reste un défi difficile, et il est souvent nécessaire de recourir à une combinaison de techniques pour obtenir un résumé de qualité.

Méthodes :

1- Les réseaux de neurones récurrents (RNN)

Les réseaux de neurones récurrents (RNN) sont une architecture de réseau de neurones qui utilise des boucles pour traiter des données séquentielles, comme du texte. Dans le cas de la synthèse automatique de texte, un RNN prend en entrée un certain nombre de mots précédemment générés et essaie de prédire le mot suivant. Pour cela, chaque mot est transformé en un vecteur de caractéristiques (appelé "embedding") qui est utilisé comme entrée dans le RNN. Le RNN comprend alors plusieurs couches de neurones, chacune possédant une mémoire interne appelée "état caché".

Lorsque le RNN traite chaque mot, il utilise l'état caché précédent et le vecteur d'embedding du mot actuel pour mettre à jour l'état caché et prédire le mot suivant. Le RNN utilise également un processus de "backpropagation through time" pour entraîner ses poids et ainsi améliorer sa précision de prédiction. Cela implique de calculer les erreurs de prédiction sur chaque mot et de les utiliser pour mettre à jour les poids du RNN afin de réduire ces erreurs.

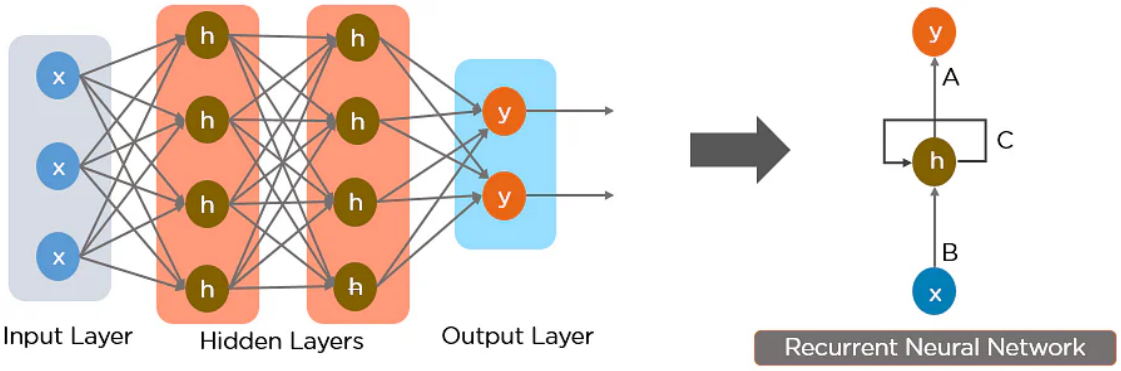

figure 1 : RNN simple

Voici quelques avantages potentiels des RNN dans la synthèse de texte:

- Capacité à traiter des séquences de données de longueur variable: les RNN sont capables de traiter des séquences de données de longueur variable, ce qui est utile dans le cadre de la synthèse de texte, où les textes peuvent avoir des longueurs très différentes.
- Capacité à traiter le contexte et les relations séquentielles entre les mots: les RNN sont capables de traiter le contexte et les relations séquentielles entre les mots d'un texte, ce qui est important pour comprendre le sens des phrases et des paragraphes.
- Capacité à utiliser des données de traitement du langage naturel préalablement étiquetées: les RNN peuvent être entrainées à l'aide de données de traitement du langage naturel préalablement étiquetées, ce qui peut être utile pour la synthèse de texte car il est souvent difficile de trouver des données d'entraînement étiquetées de manière appropriée.

Voici quelques limitations potentielles des RNN dans la synthèse de texte:

- Nécessité de disposer de grandes quantités de données d'entraînement: pour que les RNN fonctionnent efficacement, ils ont généralement besoin de grandes quantités de données d'entraînement, ce qui peut être un défi dans le cas de la synthèse de texte, où il peut être difficile de trouver de grandes quantités de données d'entraînement étiquetées de manière appropriée.
- Complexité de la formation et de l'entraînement: les RNN peuvent être complexes à mettre en place et à entraîner, ce qui peut être un défi pour les utilisateurs qui ne sont pas familiers avec ces modèles.
- Difficultés à capturer les relations à long terme: les RNN peuvent avoir du mal à capturer les relations à long terme dans un texte, ce qui peut être un problème pour la synthèse de texte, où il est souvent important de comprendre les relations à long terme entre les différentes parties du texte.

2 - GRU

L'architecture de GRU (Gated Recurrent Unit) est un modèle de réseau de neurones récurrent qui a été conçu pour la synthèse automatique de textes. Elle est particulièrement adaptée pour traiter des données séquentielles, telles que des phrases ou des paragraphes, car elle permet de conserver les informations précédemment entrées dans le réseau. La structure de base d'un GRU est constituée de cellules de mémoire récurrentes, qui sont reliées les unes aux autres dans un réseau en chaîne. Chaque cellule de mémoire est responsable de stocker une portion de l'information de la séquence, en utilisant une fonction de mémoire qui permet de mettre à jour et de conserver les informations précédemment entrées. Le GRU utilise également des portes de mémoire, qui sont des fonctions de filtrage qui permettent de contrôler l'accès à l'information stockée dans chaque cellule de mémoire. Ces portes de mémoire sont contrôlées par des poids de réseau qui sont mis à jour à chaque étape de la séquence, permettant ainsi de choisir quelles informations sont conservées et mises à jour et lesquelles sont oubliées. En utilisant cette architecture de mémoire récurrente et de portes de mémoire, le GRU est capable de traiter des séquences de données de manière efficace et de synthétiser du texte en utilisant les informations précédemment entrées dans le réseau. Cette approche est particulièrement utile pour la synthèse de textes, car elle permet de préserver les informations contextuelles et de créer du texte qui est cohérent et fluide.

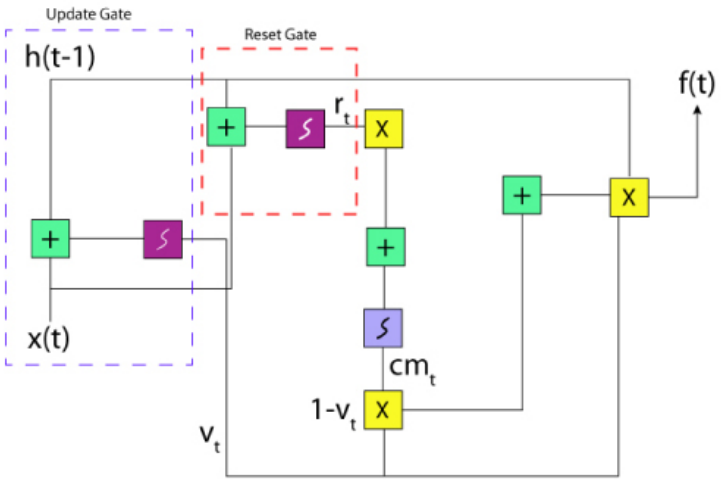

figure 2 : *L'architecture de travail de Gated Recurrent Unit (GRU)*

Voici quelques avantages potentiels des GRU dans la synthèse de texte:

- Simplicité de mise en place et d'entraînement: les GRU sont plus simples à mettre en place et à entraîner que les RNN traditionnelles, ce qui peut être un avantage pour les utilisateurs qui ne sont pas familiers avec les RNN.
- Capacité à traiter des séquences de données de longueur variable: comme les RNN, les GRU sont capables de traiter des séquences de données de longueur variable, ce qui est utile dans le cadre de la synthèse de texte, où les textes peuvent avoir des longueurs très différentes.

- Capacité à traiter le contexte et les relations séquentielles entre les mots: les GRU sont capables de traiter le contexte et les relations séquentielles entre les mots d'un texte, ce qui est important pour comprendre le sens des phrases et des paragraphes.

Voici quelques limitations potentielles des GRU dans la synthèse de texte:

- Moins de flexibilité que les RNN traditionnelles: les GRU sont moins flexibles que les RNN traditionnelles car ils ont moins de paramètres, ce qui peut limiter leur capacité à apprendre certaines relations complexes dans le texte.
- Nécessité de disposer de grandes quantités de données d'entraînement: comme les RNN, les GRU ont généralement besoin de grandes quantités de données d'entraînement pour fonctionner efficacement, ce qui peut être un défi dans le cas de la synthèse de texte, où il peut être difficile de trouver de grandes quantités de données d'entraînement étiquetées de manière appropriée.

3 - LSTM

Les unités de mémoire à long terme (LSTM) sont un type de réseau de neurones récurrents (RNN) qui a été conçu pour améliorer la capacité des RNN à traiter les séquences de données à long terme. Les LSTM utilisent des portes de mémoire et des cellules de mémoire pour contrôler l'accès et la rétention de l'information à long terme, ce qui les rend particulièrement adaptées aux tâches de traitement du langage naturel qui nécessitent de comprendre les relations à long terme dans un texte

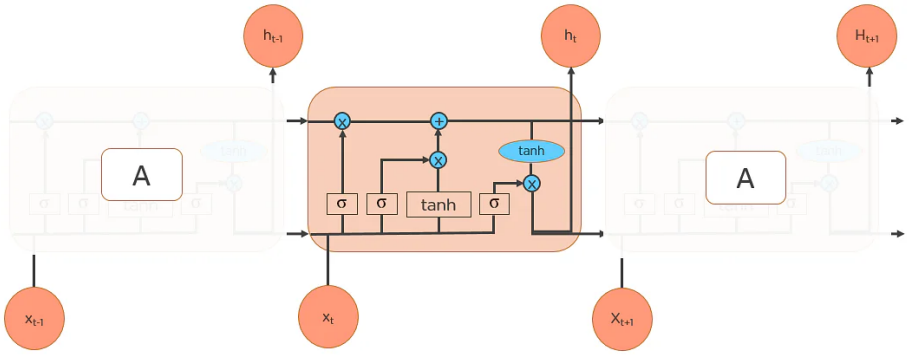

figure 3 : Structure LTSM

Voici quelques avantages potentiels des LSTM dans la synthèse de texte:

- Capacité à traiter des séquences de données à long terme: les LSTM sont particulièrement adaptées pour traiter les séquences de données à long terme, ce qui peut être utile dans le cadre de la synthèse de texte, où il est souvent important de comprendre les relations à long terme entre les différentes parties du texte.
- Capacité à traiter le contexte et les relations séquentielles entre les mots: comme les RNN et les GRU, les LSTM sont capables de traiter le contexte et les relations séquentielles entre les mots d'un texte, ce qui est important pour comprendre le sens des phrases et des paragraphes.
- Capacité à utiliser des données de traitement du langage naturel préalablement étiquetées: les LSTM peuvent être entraînées à l'aide de données de traitement du langage naturel préalablement étiquetées, ce qui peut être utile pour la synthèse de texte car il est souvent difficile de trouver des données d'entraînement étiquetées de manière appropriée.

Voici quelques limitations potentielles des LSTM dans la synthèse de texte:

- Complexité de la formation et de l'entraînement: les LSTM peuvent être complexes à mettre en place et à entraîner, ce qui peut être un défi pour les utilisateurs qui ne sont pas familiers avec ces modèles.
- Nécessité de disposer de grandes quantités de données d'entraînement: pour que les LSTM fonctionnent efficacement, elles ont généralement besoin de grandes quantités de données d'entraînement, ce qui peut être un défi dans le cas de la synthèse de texte, où il peut être difficile de trouver de grandes quantités de données d'entraînement étiquetées de manière appropriée.

4- Transformers

Les transformateurs sont devenus un choix populaire pour la synthèse de texte en raison de leur succès dans les tâches de traitement du langage naturel telles que la traduction automatique et la modélisation du langage.

Les origines des transformateurs remontent aux années 1980 avec le développement du mécanisme d'auto-attention, qui permet au modèle de considérer le contexte et les dépendances de chaque mot dans une phrase. En 2014, Vaswani et al. introduit le modèle Transformer, qui utilise l'auto-attention pour analyser et traiter de longues séquences de texte sans avoir besoin de réseaux de neurones récurrents (RNN). Cela a rendu le modèle plus efficace et plus rapide à former, ce qui a conduit à son adoption généralisée dans les tâches NLP. En 2018, le modèle Transformer a encore été amélioré avec l'introduction du modèle BERT (Représentations d'encodeurs bidirectionnels de transformateurs). BERT a pu obtenir des résultats de pointe

sur un large éventail de tâches NLP, y compris la synthèse de texte. Cela a encore renforcé l'importance des transformateurs dans la PNL et a conduit au développement d'autres modèles basés sur les transformateurs tels que GPT (Generative Pre-training Transformer) et RoBERTa (Robustly Optimized BERT). Dans l'ensemble, le succès des transformateurs dans les tâches NLP, en particulier la synthèse de texte, peut être attribué à leur capacité à analyser et à traiter efficacement de longues séquences de texte sans avoir besoin de RNN, ce qui les rend plus efficaces et plus rapides à former.

Le modèle Transformer extrait les caractéristiques de chaque mot à l'aide d'un mécanisme d'auto-attention pour déterminer l'importance de tous les autres mots de la phrase. au mot précité. Et aucune unité récurrente n'est utilisée pour obtenir ces fonctionnalités, ce ne sont que des sommes pondérées et des activations, elles peuvent donc être très parallélisables et efficaces.

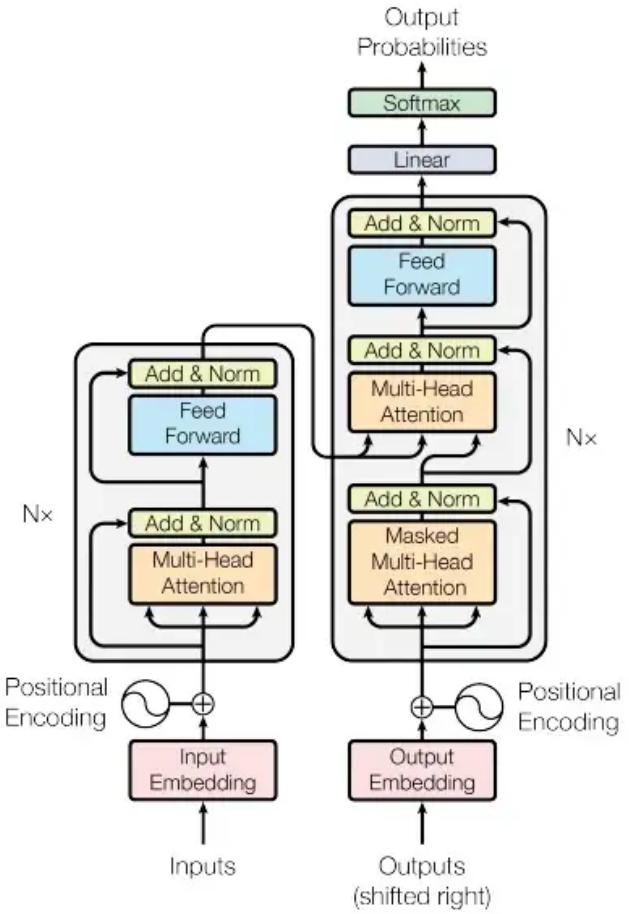

figure 4 : Extrait de l'article "Attention is all you need" de Vaswani, et al., 2017

Auto-attention :

Une fonction d'attention peut être décrite comme mappant une requête et un ensemble de paires clé-valeur 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 des valeurs, où le poids attribué à chaque valeur est calculé par une fonction de compatibilité de la requête avec la clé correspondante

Chaque vecteur d'entrée est utilisé de trois manières différentes dans le mécanisme d'auto-attention : la requête, la clé et la valeur. Dans chaque rôle, il est comparé aux autres vecteurs pour obtenir sa propre sortie yi(Query), pour obtenir la jième sortie yj(Key) et pour calculer chaque vecteur de sortie une fois les poids établis (Value). Pour obtenir ces rôles, nous avons besoin de trois matrices de poids de dimensions k*k et calculons trois transformations linéaires pour chaque xi :

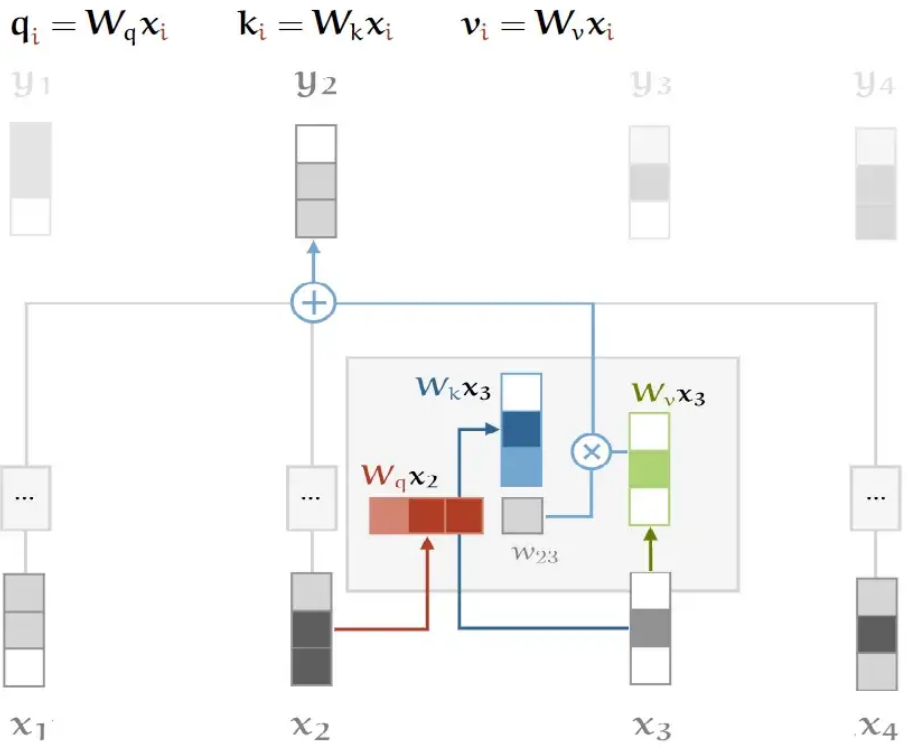

Illustration of the self-attention with key, query and value

figure 5 : Extrait de "transformers from scratch" par Peter Bloem

Ces trois matrices sont généralement appelées K, Q et V, trois couches de poids apprenables qui sont appliquées à la même entrée codée. Par conséquent, comme chacune de ces trois matrices provient de la même entrée, on peut appliquer le mécanisme d'attention du vecteur d'entrée avec lui-même, une « auto-attention ».

Attention mise à l'échelle du produit scalaire (Scaled Dot Product Attention)

"L'entrée se compose de requêtes q et de clés de dimension k , et de valeurs de dimension v . Nous calculons les produits scalaires de la requête avec toutes les clés, divisons chacun par sqrt(dk) et appliquons une fonction softmax pour obtenir les poids sur les valeurs." - papier "Attention is all you need"

Ensuite, nous utilisons les matrices Q, K et V pour calculer les scores d'attention. Les scores mesurent la concentration à accorder à d'autres endroits ou mots de la séquence d'entrée par rapport à un mot à une certaine position. C'est-à-dire le produit scalaire du vecteur de requête avec le vecteur clé du mot respectif que nous évaluons. Ainsi, pour la position 1, nous calculons le produit scalaire (.) de q1 et k1, puis q1*k2, q1*k3 et ainsi de suite…

Ensuite, nous appliquons le facteur "mise à l'échelle" pour avoir des gradients plus stables. La fonction softmax ne peut pas fonctionner correctement avec de grandes valeurs, ce qui entraîne la disparition des gradients et ralentit l'apprentissage. Après "softmaxing", nous multiplions par la matrice de valeurs pour conserver les valeurs des mots sur lesquels nous voulons nous concentrer et minimisons ou supprimons les valeurs des mots non pertinents (sa valeur dans la matrice V doit être très petite).

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

figure 6 : Scaled Dot Product Attention

Attention multi-tête

Dans la description précédente, les scores d'attention sont concentrés sur la phrase entière à la fois, cela produirait les mêmes résultats même si deux phrases contiennent les mêmes mots dans un ordre différent. Au lieu de cela, nous aimerions nous occuper de différents segments des mots

"Nous pouvons donner à l'attention de soi un plus grand pouvoir de discrimination, en combinant plusieurs têtes d'attention de soi, en divisant les vecteurs de mots en un nombre fixe (h, nombre de têtes) de morceaux, puis l'attention de soi est appliquée sur les morceaux correspondants, en utilisant Sous-matrices Q, K et V."

- Peter Bloem, Transformers from scratch

Mais la couche suivante (la couche Feed-Forward) attend une seule matrice, un vecteur pour chaque mot, donc "après avoir calculé le produit scalaire de chaque tête, nous concaténons les matrices de sortie et les multiplions par une matrice de poids supplémentaire Wo". Cette matrice finale capture les informations de toutes les têtes d'attention.

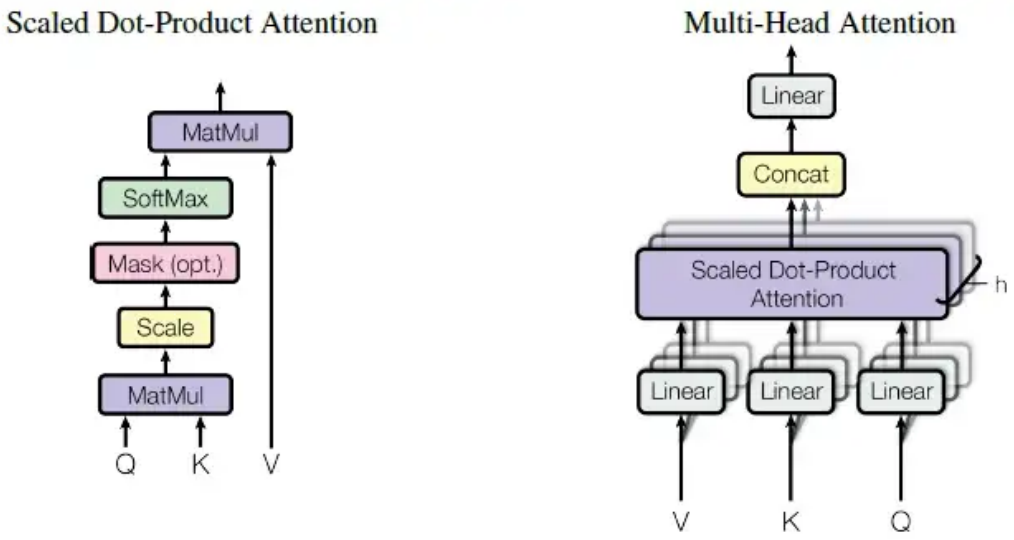

figure 7 : Extrait de l'article "Attention is all you need" de Vaswani, et al., 2017

Encodage positionnel

Nous avons mentionné brièvement que l'ordre des mots dans la phrase est un problème à résoudre dans ce modèle, car le réseau et le mécanisme d'auto-attention sont invariants par permutation. Si nous mélangeons les mots dans la phrase d'entrée, nous obtenons les mêmes solutions. Nous devons créer une représentation de la position du mot dans la phrase et l'ajouter au mot incorporation.

Ainsi, nous appliquons une fonction pour mapper la position dans la phrase à un vecteur à valeur réelle. Le réseau apprendra à utiliser ces informations. Une autre approche consisterait à utiliser une position intégrée, similaire à l'intégration de mots, codant chaque position connue avec un vecteur.

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

figure 8 : Encodage positionnel

Maintenant que toutes les pièces principales du modèle ont été décrites, nous pouvons introduire les composants du codeur.

Encodeur

- Encodage positionnel : Ajoutez l'encodage de position à l'intégration d'entrée (nos mots d'entrée sont transformés en vecteurs d'intégration).
- N = 6 couches identiques, contenant deux sous-couches : un mécanisme d'auto-attention multi-têtes et un réseau d'anticipation entièrement connecté (deux transformations linéaires avec une activation ReLU). Mais il est appliqué en position à l'entrée, ce qui signifie que le même réseau de neurones est appliqué à chaque vecteur « jeton » appartenant à la séquence de phrases.
- Il existe une connexion résiduelle autour de chaque sous-couche (attention et réseau FC), résumant la sortie de la couche avec son entrée, suivie d'une normalisation de couche.
- Avant chaque connexion résiduelle, une régularisation est appliquée : « Nous appliquons le dropout à la sortie de chaque sous-couche, avant qu'il ne soit ajouté à l'entrée de la sous-couche et normalisé. De plus, nous appliquons l'abandon aux sommes des plongements et des encodages positionnels dans les piles d'encodeur et de décodeur » [1] avec un taux d'abandon de 0,1.

Décodeur

- Encodage positionnel : similaire à celui de l'encodeur
- N=6 couches identiques, contenant 3 trois sous-couches. Tout d'abord, l'attention masquée multi-tête ou l'attention causale masquée pour empêcher les positions d'assister aux positions suivantes. Il est implémenté en fixant à $-\infty$ les valeurs correspondant aux états interdits dans la couche softmax des modules d'attention des produits scalaires. Le deuxième composant ou « attention de l'encodeur-décodeur » effectue une attention multi-tête sur la sortie du décodeur, les vecteurs clé et valeur proviennent de la sortie de l'encodeur mais les requêtes proviennent de la couche de décodeur précédente. "Cela permet à chaque position dans le décodeur d'être présente sur toutes les positions de la séquence d'entrée" [1]. Et enfin le réseau est entièrement connecté.
- La connexion résiduelle et la normalisation de couche autour de chaque sous-couche, similaire à l'encodeur.
- Et répétez le même **dropout** résiduel qui a été exécuté dans l'encodeur.

Une fois que nous avons défini nos composants et créé l'encodeur, le décodeur et la couche finale linear-softmax, nous assemblons les pièces pour former notre modèle, le Transformer.

Il est à noter que nous créons 3 masques dont chacun nous permettra :

- **Masque d'encodeur :** il s'agit d'un masque de remplissage pour éliminer les jetons de remplissage du calcul de l'attention.

- **Masque décodeur 1 :** ce masque est une union du masque de remplissage et du masque d'anticipation qui aidera l'attention causale à rejeter les jetons « dans le futur ». Nous prenons la valeur maximale entre le masque de remplissage et celui d'anticipation.
- **Masque de décodeur 2 :** c'est le masque de remplissage et il est appliqué dans la couche d'attention de l'encodeur-décodeur.

5 - Apprentissage par transfert (Text to text transfer Transformer - t5)

Pourquoi ?

Nous disposons de peu de données annotées pour cette tâche qui nécessite normalement une grande base de données, alors c'est mieux d'utiliser le modèle pré-entraîné qui a déjà appris à extraire des caractéristiques utiles des données.

Il existe plusieurs raisons pour lesquelles l'utilisation du transformateur T5 peut être un meilleur choix pour la synthèse de texte que la création d'un transformateur à partir de rien :

- **Performance :** T5 a été formé sur une quantité massive de données et a obtenu des résultats de pointe sur une variété de tâches, y compris la synthèse de texte. Cela signifie qu'il est susceptible de mieux fonctionner sur votre tâche de résumé de texte qu'un simple transformateur que vous pourriez créer à partir de zéro.
- **Facilité d'utilisation :** T5 est un modèle pré-formé qui peut être facilement ajusté pour le résumé de texte, ce qui signifie que vous n'avez pas à passer beaucoup de temps et d'efforts à créer et à former un modèle à partir de zéro.
- **Apprentissage par transfert :** en utilisant un modèle pré-formé comme T5, vous pouvez tirer parti des connaissances que le modèle a apprises à partir d'un large éventail de tâches et les appliquer à votre tâche de synthèse de texte. Cela peut vous aider à obtenir de meilleurs résultats, surtout si vous disposez d'une quantité limitée de données d'entraînement.
- **Support communautaire :** T5 est un modèle activement recherché, et il existe une grande communauté de chercheurs et de développeurs qui y travaillent. Cela signifie que vous pouvez bénéficier des connaissances et de l'expérience collectives de cette communauté, ainsi que de toutes les mises à jour ou améliorations du modèle publiées.

Dans l'ensemble, l'utilisation d'un modèle pré-formé comme T5 peut être un bon choix pour le résumé de texte car il peut vous faire gagner du temps et des efforts, et il peut également vous aider à obtenir de meilleurs résultats en tirant parti des connaissances et de l'expertise de la communauté.

Explication des données :

Le BBC News Summary dataset est une collection d'articles d'actualité et de résumés du site Web de BBC News. Il comprend des articles de diverses catégories telles que la politique, les sports, la technologie, le divertissement et la technologie. Chaque article de l'ensemble de données contient les renseignements suivants :

- Titre de l'article
- Le texte intégral de l'article
- Résumé de l'article

Cet ensemble de données contient quatre cent dix-sept articles d'actualité politique BBC de 2004 à 2005 dans le dossier 'News Articles', ces articles sont généralement d'environ 500-1000 mots, la première clause du texte des articles est le titre respectif.
Pour chaque article, un résumé est fourni dans le dossier 'Summaries' qui est une version plus courte de l'article qui transmet les principaux points.

Aperçu sur l'ensemble de données :

	original	summary
0	Ad sales boost Time Warner profit Quarterly profits at US media giant TimeWarner jumped 76% to $1.13bn (Â£600m) for the three months to December, from $639m year-earlier. The firm, which is now one of the biggest investors in Google, benefited from sales of high-speed internet connections and higher advert sales. TimeWarner said fourth quarter sales rose 2% to $11.1bn from $10.9bn. Its profits were buoyed by one-off gains which offset a profit dip at Warner Bros, and less users for AOL. Time Warner said on Friday that it now owns 8% of search-engine Google. But its own internet business, AOL, had has mixed fortunes. It lost 464,000 subscribers in the fourth quarter profits were lower than in the preceding three quarters. However, the company said AOL's underlying profit before exceptional items rose 8% on the back of stronger internet advertising revenues. It hopes to increase subscribers by offering the online service free to TimeWarner internet customers and will try to sign up AOL's existing customers for high-speed broadband. TimeWarner also has to restate 2000 and 2003 results following a probe by the US Securities Exchange Commission (SEC), which is close to concluding. Time Warner's fourth quarter profits were slightly better than analysts' expectations. But its film division saw profits slump 27% to $284m, helped by box-office flops Alexander and Catwoman, a sharp contrast to year-earlier, when the third and final film in the Lord of the Rings trilogy boosted results. For the full-year, TimeWarner posted a profit of $3.36bn, up 27% from its 2003 performance, while revenues grew 6.4% to $42.09bn. "Our financial performance was strong, meeting or exceeding all of our full-year objectives and greatly enhancing our flexibility," chairman and chief executive Richard Parsons said. For 2005, TimeWarner is projecting operating earnings growth of around 5%, and also expects higher revenue and wider profit margins.	TimeWarner said fourth quarter sales rose 2% to $11.1bn from $10.9bn.For the full-year, TimeWarner posted a profit of $3.36bn, up 27% from its 2003 performance, while revenues grew 6.4% to $42.09bn.Quarterly profits at US media giant TimeWarner jumped 76% to $1.13bn (Â£600m) for the three months to December, from $639m year-earlier.However, the company said AOL's underlying profit before exceptional items rose 8% on the back of stronger internet advertising revenues.Its profits were buoyed by one-off gains which offset a profit dip at Warner Bros, and less users for AOL.For 2005, TimeWarner is projecting operating earnings growth of around 5%, and also expects higher revenue and wider profit margins.It lost 464,000 subscribers in the fourth quarter profits were lower than in the preceding three quarters.Time Warner's fourth quarter profits were slightly better than analysts' expectations.

Implémentation et résultats :

Dans cette section, nous présenterons les performances obtenues après l'application de ces différentes méthodes de mise à l'échelle. Notre approche consiste à tester le modèle t5, puis à recycler le modèle sur notre ensemble de données et à vérifier les résultats.
Nous avons téléchargé le modèle t5 pré-entraîné (t5-small) et avant d'entraîner le modèle sur notre ensemble de données pour la tâche d'apprentissage profond supervisé, nous avons tokenisé les entrées et généré les résumés, bien sûr, nous avons d'abord dû prétraiter les données en supprimant les espaces, lignes et tabulations supplémentaires. Pour le reste, les données sont assez propres.
Il y a plusieurs raisons pour lesquelles il est difficile de calculer la performance du modèle pour la synthèse, et le plus important est que la qualité du texte synthétisé dépend de nombreux facteurs, tels que la grammaire, la syntaxe, la cohérence et la pertinence du contenu. Il est donc difficile de mesurer la performance du modèle de manière objective.
Prenant ça en considération, nous avons calculé la métrique Rouge qui mesure le rappel : combien de mots (et/ou n-grammes) dans les résumés de référence humains sont apparus dans les résumés générés par la machine.

Rouge score	0.431

Note :
Il y a une autre chose à prendre en considération, c'est que les longueurs des résumés prédites sont différentes par rapport au résumés originaux fournis dans le dataset.

Exemple de prédiction :

Sommaire original :

> Rod Eddington, BA's chief executive, said the results were "respectable" in a third quarter when fuel costs rose by £106m or 47.3%.To help offset the increased price of aviation fuel, BA last year introduced a fuel surcharge for passengers.BA had previously forecast a 2% to 3% rise in full-year revenue."It is quite good on the revenue side and it shows the impact of fuel surcharges and a positive cargo development, however, operating margins down and cost impact of fuel are very strong," he said.Yet aviation analyst Mike Powell of Dresdner Kleinwort Wasserstein says BA's estimated annual surcharge revenues - £160m - will still be way short of its additional fuel costs - a predicted extra £250m."For the year to March 2005, the total revenue outlook is slightly better than previous guidance with a 3% to 3.5% improvement anticipated," BA chairman Martin Broughton said.Looking ahead to its full year results to March 2005, BA warned that yields - average revenues per passenger - were expected to decline as it continues to lower prices in the face of competition from low-cost carriers.BA's profits were still better than market expectation of £59m, and it expects a rise in full-year revenues.

Sommaire prédit :

> the airline made a pre-tax profit of £75m ($141m) compared with £125m a year earlier. the results were "respectable" in a third quarter when fuel costs rose by £106m or 47.3%. it expects a rise in full-year revenues to offset the increased price of aviation fuel. the airline had previously forecast a 2% to 3% rise in full-year revenue.

On peut notamment voir que les longueurs sont différentes mais le contexte reste le même et le sommaire prédit peut être plus précis.

Notre reflection :

Les résultats sont très bons en termes de performances compte tenu des circonstances. Il y a bien sûr des améliorations possibles en utilisant le transfer learning qui va consister des étapes suivantes :

- Tokenisation de la dataset :
- Définition des paramètres initiaux de l'apprentissage : Cette phase est la plus importante. Pour que le transfert d'apprentissage soit efficace, il est important de spécifier correctement les paramètres d'apprentissage, tels que le taux d'apprentissage et la régularisation. Ces paramètres ont un impact sur la façon dont le modèle s'ajuste aux données de la nouvelle tâche et sur sa capacité à généraliser ses connaissances à de nouvelles données. Si les paramètres d'apprentissage ne sont pas correctement spécifiés, le modèle pourrait ne pas être en mesure de tirer pleinement parti des connaissances acquises lors de son pré-entraînement, ce qui peut entraîner une performance inférieure sur la nouvelle tâche.

Le problème majeur que nous avons rencontré était la définition correcte des paramètres car il faut tester les paramètres à chaque itération et c'est très difficile lorsque le modèle prend plus que 6 heures pour faire une seule époque.

Pour conclure, nous pensons que cette approche sera efficace mais avec beaucoup de temps et de puissance de calcul que nous n'avons pas actuellement

Conclusion:

La synthèse de texte consiste à résumer un texte en utilisant les informations les plus importantes et essentielles.
Le modèle T5 (Text-To-Text Transfer Transformer) est un modèle de traitement du langage naturel développé par Google qui peut être utilisé pour la synthèse de texte.
Le modèle T5 a été entraîné sur un très grand ensemble de données et est capable de réaliser de nombreuses tâches de traitement du langage naturel, y compris la synthèse de texte. Il utilise une architecture de transformateur pour encoder le texte d'entrée et générer du texte de sortie en utilisant l'attention de transformer.
Le modèle T5 a été très performant dans plusieurs tâches de traitement du langage naturel, notamment la synthèse de texte. Cependant, comme pour tout modèle de traitement du langage naturel, il peut y avoir des limites à sa performance et il est important de le tester et de le valider avant de l'utiliser en production.
En résumé, le modèle T5 est un outil puissant pour la synthèse de texte, mais il est important de le tester et de le valider avant de l'utiliser en production.

Références :

https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn

https://www.kaggle.com/datasets/pariza/bbc-news-summary

https://papers.neurips.cc/paper/7181-attention-is-all-you-need.pdf

https://towardsdatascience.com/attention-is-all-you-need-discovering-the-transformer-paper-73e5ff5e0634

https://peterbloem.nl/blog/transformers