Dénombrement et combinatoire

Pourquoi lire un texte long coûte si cher à une IA

À savoir avant de commencer

Ce document réutilise le document 2 (vecteurs de mots, produit scalaire) : le mécanisme d'attention, qui compare chaque mot à chaque autre mot du contexte, est un exemple réel et important d'application du dénombrement.

Niveau : Terminale (spécialité mathématiques)

Introduction : comparer chaque mot à chaque autre mot

Dans le document 2, on a vu qu'une IA compare des mots entre eux à l'aide du produit scalaire de leurs vecteurs. Le mécanisme d'attention, au cœur des modèles de langage actuels, généralise cette idée à un texte entier : pour bien interpréter un mot, le modèle le compare à tous les autres mots du contexte (pas seulement au précédent, comme dans le document 1).

La question de ce document : combien de comparaisons cela représente-t-il, et pourquoi ce nombre devient-il vite un problème ?

Partie A — Rappels de dénombrement

Niveau : Terminale

Rappels de cours

Principe multiplicatif : si un premier choix offre p possibilités et qu'un second choix, indépendant du premier, en offre q, alors il y a $p \times q$ façons de faire les deux choix successivement.

Permutations : le nombre de façons d'ordonner n objets distincts est $n! = n \times (n-1) \times \dots \times 2 \times 1$ (factorielle de n).

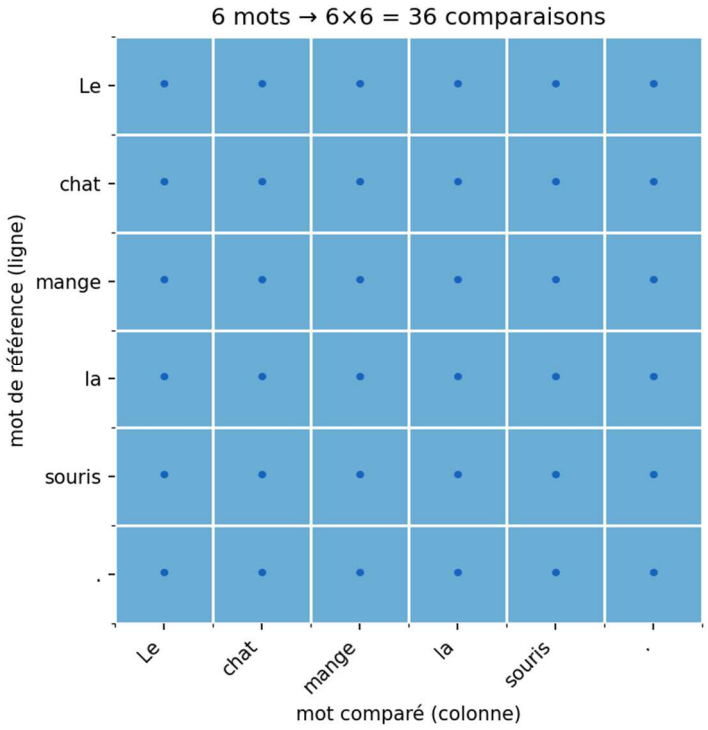
Combinaisons : le nombre de façons de choisir k objets parmi n, sans tenir compte de l'ordre, est $C(n,k) = n! / (k! \times (n-k)!)$.

1. Une bibliothèque a 5 livres différents. De combien de façons peut-on les ranger côte à côte sur une étagère ?
2. Dans un groupe de 6 amis, de combien de façons peut-on choisir une équipe de 2 personnes pour un jeu (l'ordre n'a pas d'importance) ?
3. Toujours avec 6 amis, de combien de façons peut-on choisir un binôme ordonné (un « premier » et un « second », par exemple pour un relais) ?

Partie B — Le coût du mécanisme d'attention

Niveau : Terminale

Pour interpréter chaque mot d'une phrase de n mots, le mécanisme d'attention calcule un score de similarité (document 2) entre chaque mot et chaque autre mot du texte — y compris lui-même.



4. Pour une phrase de n mots, combien de paires (mot de référence, mot comparé) doit-on former, si l'on compte chaque mot comparé à chaque autre (y compris à lui-même), et si l'ordre compte (comparer « chat » à « souris » n'est pas la même opération que comparer « souris » à « chat », car ce sont deux calculs distincts) ? Justifie à l'aide du principe multiplicatif.
5. Vérifie ta formule sur l'exemple du graphique ci-dessus (n=6 mots) : combien de points obtiens-tu ?

Complète le tableau suivant, qui donne le nombre de comparaisons nécessaires selon la taille du texte :

n (nombre de mots)	10	100	1 000	100 000
n^2 comparaisons	?	?	?	?

6. D'après ce tableau, que se passe-t-il lorsque la taille du texte est multipliée par 10 ? Le nombre de comparaisons est-il aussi multiplié par 10 ?

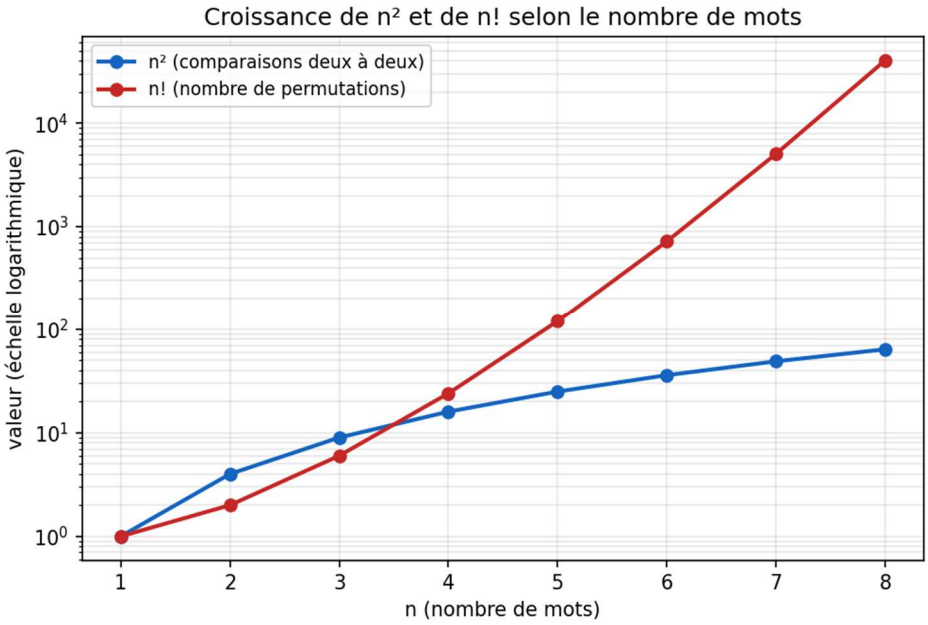
Partie C — Pourquoi l'ordre des mots doit être ajouté à part

Niveau : Terminale

Le mécanisme d'attention, tel que décrit en partie B, calcule un score entre chaque paire de mots à partir de leur contenu (leurs vecteurs, document 2) — mais rien, dans ce calcul, ne dépend de la position des mots dans la phrase. Si l'on mélangeait complètement l'ordre des mots d'une phrase, le mécanisme d'attention « brut » calculerait exactement les mêmes scores entre les mêmes paires de mots.

7. Combien y a-t-il de façons différentes d'ordonner les n=6 mots d'une phrase (utilise la partie A) ?
8. Compare ce nombre à celui obtenu en B.2 (n^2 pour n=6). Lequel des deux nombres est le plus grand ?

Le graphique suivant compare la croissance de n^2 et de n! (échelle logarithmique, à cause de l'explosion de n!) :



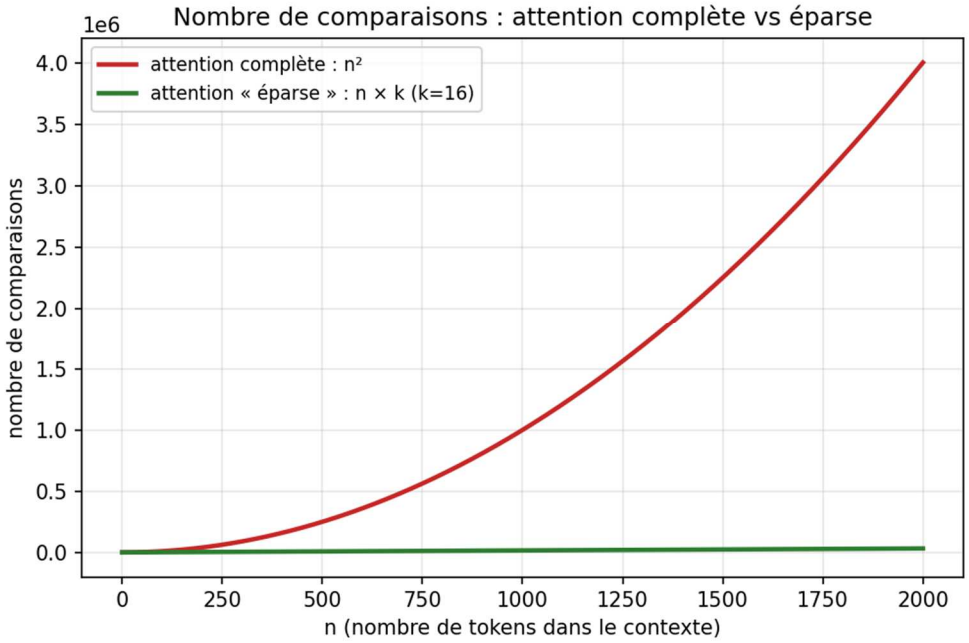
9. D'après le graphique, à partir de combien de mots environ n! dépasse-t-il clairement n² ? Que peux-tu en dire sur la vitesse de croissance comparée de ces deux grandeurs ?

C'est précisément parce que le calcul brut du produit scalaire entre vecteurs ignore l'ordre des mots que les IA doivent ajouter une information de position à chaque vecteur de mot (document 2), appelée encodage positionnel, pour que le modèle sache distinguer « le chat mange la souris » de « la souris mange le chat ».

Partie D — Réduire le nombre de comparaisons : l'attention « éparse »

Niveau : Terminale

Pour des textes très longs, le coût en n² devient vite excessif. Une idée utilisée par certains modèles récents consiste à ne comparer chaque mot qu'à un nombre limité k d'autres mots (par exemple, les k mots les plus proches dans le texte), plutôt qu'à tous : on parle d'attention éparse (« sparse attention », en anglais).



10. Pour un texte de n mots, où chaque mot est comparé à seulement k=16 autres mots (au lieu des n−1 autres mots), combien de comparaisons cela représente-t-il au total, en fonction de n et de k ? (Justifie à l'aide du principe multiplicatif.)
11. Pour n=1 000 mots, calcule le nombre de comparaisons en attention complète (n²) et en attention éparse (n×16). Par quel facteur le nombre de comparaisons a-t-il été réduit ?

12. D'après le graphique, pourquoi l'écart entre les deux courbes se creuse-t-il de plus en plus lorsque n augmente ?

Ce que ce modèle simplifie, et ce que fait réellement une IA comme Claude

Ce que ce modèle simplifie	Ce que fait réellement un modèle comme Claude
Un décompte simple de n^2 comparaisons, présenté comme un problème purement combinatoire. Une attention éparse décrite comme « chaque mot regarde ses k voisins », de façon uniforme. Un encodage positionnel présenté brièvement, comme un simple correctif.	En réalité, chaque comparaison n'est pas un simple produit scalaire, mais un calcul matriciel complet (projections des vecteurs par des matrices apprises), et un modèle empile des dizaines de couches d'attention, chacune répétant ce coût en n^2 — ce qui rend le coût total du calcul l'un des principaux facteurs limitants des IA actuelles sur de très longs textes. Les schémas réels d'attention éparse (Longformer, BigBird, attention glissante...) sont plus sophistiqués : ils combinent des motifs de comparaisons locales, globales et parfois aléatoires, choisis pour préserver le plus possible la qualité du modèle tout en réduisant le coût. Les encodages positionnels réellement utilisés (comme RoPE, les encodages rotatifs) sont des constructions mathématiques élaborées, bien plus riches qu'un simple numéro de position ajouté au vecteur.

Le principe central — un coût de comparaison qui croît en n^2, et la nécessité d'ajouter une information de position — reste, lui, exactement la réalité du fonctionnement des modèles de langage actuels, et un sujet de recherche à part entière pour le rendre plus efficace.

Exercices d'entraînement

Exercice 1 — Paires ordonnées et non ordonnées

Niveau : Terminale

On considère une phrase de 6 mots.

13. Combien de paires ordonnées (mot de référence, mot comparé), y compris les paires d'un mot avec lui-même, peut-on former ?
14. Combien de paires ordonnées peut-on former si l'on exclut les paires d'un mot avec lui-même ?
15. Combien de paires non ordonnées (au sens des combinaisons, sans répétition) de mots distincts peut-on former ?

Exercice 2 — Permutations d'un poème

Niveau : Terminale

Un poème est composé de 8 mots distincts.

16. Combien d'ordres différents peut-on former avec ces 8 mots ?
17. Combien de paires ordonnées (y compris avec répétition) peut-on former avec ces mêmes 8 mots ?
18. Compare les deux résultats. Qu'en conclus-tu sur la rapidité de croissance du nombre de permutations par rapport au nombre de paires ?

Exercice 3 — Gain de l'attention éparse

Niveau : Terminale

On considère un très long document de n=50 000 mots, traité par un modèle utilisant une attention éparse où chaque mot n'est comparé qu'à k=32 autres mots.

19. Calcule le nombre de comparaisons en attention complète (n^2), puis en attention éparse ($n \times k$).
20. Calcule le facteur de réduction (rapport entre les deux nombres obtenus).
21. Un ingénieur affirme : « avec l'attention éparse, on peut traiter des documents bien plus longs qu'avant, à coût de calcul comparable ». Explique en quoi le résultat de la question 2 appuie cette affirmation.

Corrigé

Partie A

A.1. 5! = 5×4×3×2×1 = 120 façons de ranger les 5 livres.

A.2. C(6,2) = 6!/(2!×4!) = (6×5)/(2×1) = 15 équipes possibles.

A.3. Avec un ordre (premier/second) : 6×5 = 30 binômes ordonnés possibles (principe multiplicatif : 6 choix pour le premier, puis 5 choix restants pour le second).

Partie B

B.1. Pour chaque mot de référence (n choix possibles), on peut le comparer à n'importe lequel des n mots du texte (y compris lui-même), soit n choix. Par le principe multiplicatif, le nombre total de paires est n×n = n^2.

B.2. Pour n=6 : n^2=36. On retrouve bien les 36 points de la grille 6×6 du graphique.

n (nombre de mots)	10	100	1 000	100 000
n^2 comparaisons	100	10 000	1 000 000	10 000 000 000

B.4. Lorsque n est multiplié par 10, n^2 est multiplié par 10^2=100 : le nombre de comparaisons croît beaucoup plus vite que la taille du texte elle-même (croissance quadratique, pas proportionnelle).

Partie C

C.1. Pour n=6 mots : 6! = 720 façons différentes de les ordonner.

C.2. 720 (permutations) est bien plus grand que 36 (n^2, question B.2) : le nombre de permutations dépasse largement le nombre de comparaisons deux à deux, même pour une phrase aussi courte.

C.3. D'après le graphique, n! dépasse clairement n^2 dès n=4 ou 5 environ, et l'écart devient ensuite immense (à n=8, n^2=64 alors que 8!=40 320). Le nombre de permutations croît beaucoup plus vite que n^2 : c'est une croissance factorielle, bien plus rapide qu'une croissance quadratique.

Partie D

D.1. Pour chaque mot (n choix), on le compare à seulement k autres mots choisis (k choix). Par le principe multiplicatif, le nombre total de comparaisons est n×k.

D.2. Pour n=1000 : attention complète = 1000^2 = 1 000 000 comparaisons. Attention éparse = 1000×16 = 16 000 comparaisons. Facteur de réduction = 1 000 000/16 000 = 62,5 : le nombre de comparaisons est divisé par 62,5.

D.3. n^2 croît de façon quadratique (très rapide) alors que n×k croît de façon linéaire (proportionnelle à n, puisque k est fixé) : l'écart entre les deux courbes, qui est de l'ordre de $n^2-nk = n(n-k)$, augmente donc de plus en plus vite à mesure que n grandit.

Exercice 1

1. 6^2 = 36 paires ordonnées avec répétition.

2. 6×5 = 30 paires ordonnées sans répétition (6 choix pour le premier mot, 5 choix restants pour le second).

3. C(6,2) = 15 paires non ordonnées de mots distincts.

Exercice 2

1. 8! = 40 320 ordres différents.

2. 8^2 = 64 paires ordonnées (avec répétition).

3. 40 320 est plus de 600 fois supérieur à 64 : le nombre de permutations (croissance factorielle) dépasse de très loin le nombre de paires (croissance quadratique), même pour un nombre de mots aussi modeste que 8.

Exercice 3

1. Attention complète : $50\,000^2$ = 2 500 000 000 (2,5 milliards) de comparaisons. Attention éparse : 50 000×32 = 1 600 000 (1,6 million) de comparaisons.

2. Facteur de réduction = 2 500 000 000 / 1 600 000 = 1 562,5 : le nombre de comparaisons est divisé par environ 1 562.

3. Un facteur de réduction aussi énorme (plus de 1500 fois moins de comparaisons) signifie qu'on peut traiter un document 50 000 mots avec un nombre de calculs comparable à celui d'un texte bien plus court en attention complète : cela confirme que l'attention éparse permet de traiter des textes beaucoup plus longs sans faire exploser le coût de calcul, ce qui appuie l'affirmation de l'ingénieur.

Mathématiques du programme mobilisées dans ce document

Terminale (spécialité mathématiques)

Principe multiplicatif, factorielle, permutations, combinaisons C(n,k) — l'ensemble du chapitre « Combinatoire et dénombrement » est un contenu de terminale (il n'est pas abordé en première).

Comparaison de vitesses de croissance (quadratique, factorielle, linéaire).

Lecture et interprétation de graphiques en échelle logarithmique.