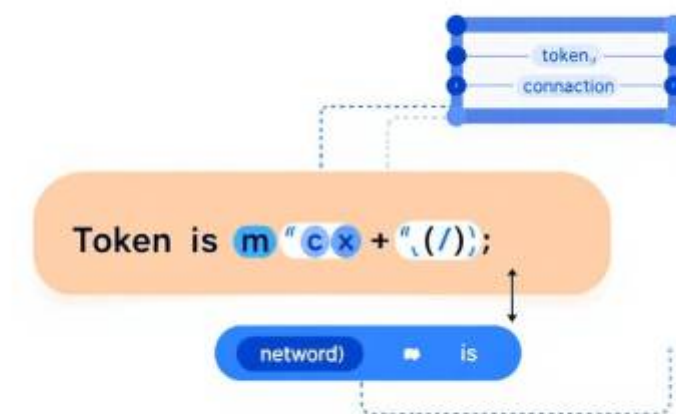


Tokens, modèles de langage et taille des données



Les **tokens** sont l'unité de base utilisée par les modèles de langage pour diviser et analyser le texte. Leur compréhension est essentielle pour optimiser l'utilisation des LLM, estimer les coûts et adapter les prompts. Cet article explique leur fonctionnement, leur lien avec la taille des données, et leur impact sur les performances des modèles comme Llama, GPT-4 ou Mistral.

□ **Difficulté :** ★★☆☆☆ (2/5)

- **Accessible :** Exemples concrets, comparaisons visuelles.
- △ **Technique :** Notions de tokenisation et d'encodage à assimiler.
- **Vocabulaire :** Token, vocabulaire, contexte, limite de tokens.

□ **Prérequis recommandés**

- **Notions de base en informatique** (octets, UTF-8).
- **Introduction au NLP** (traitement automatique du langage naturel).
- **Familiarité avec les LLM** (ex : GPT, Llama).
- **Anglais technique** basique (certains termes ne sont pas traduits).

1. Qu'est-ce qu'un token ?

1.1. Définition et exemples

- Un **token** est l'unité de base de traitement du texte par un LLM.
- Il ne correspond pas toujours à un mot entier, mais peut être :
 - Une partie de mot (*"aim"*, *"er"*).

Modèle	Limite (tokens)	Équivalent en Mo
Llama 3.2	8 000-128 000	0,03-0,5 Mo
GPT-4	128 000	0,5 Mo
Mistral	32 000	0,13 Mo
Gemini	32 000-128 000	0,13-0,5 Mo

- **Conséquence** : Un texte dépassant ces limites doit être **tronçonné** ou **résumé**.

3.2. Coût et performance

- **Facturation** : Les API facturent au token (ex : 0,001 \$/1 000 tokens).
 - **1 Mo de texte (250 000 tokens) ≈ 0,25 \$** sur certaines plateformes.
- **Vitesse** : Plus le nombre de tokens est élevé, plus le traitement est lent.

Modèle	Coût/1M tokens	Temps de traitement
Llama 3.2	~0,10 \$	Rapide
GPT-4	~0,30 \$	Modéré
Mistral	~0,15 \$	Très rapide
Gemini	~0,20 \$	Variable

4. Outils pour visualiser la tokenisation

4.1. Démonstrations en ligne

- [Tokenizer Openai](#) : Visualisez la tokenisation pour GPT.
- [Calculateur de tokens](#) : Outil interactif pour Gemini et OpenAI.

4.2. Code Python

Exemple de code Python afin d'estimer la tokenisation avec le LLM Mistral :

```
from transformers import AutoTokenizer
tokenizer = AutoTokenizer.from_pretrained("mistralai/Mistral-7B-v0.1")
tokens = tokenizer("Ton texte ici")
print(f"Nombre de tokens : {len(tokens['input_ids'])}")
```

5. Bonnes pratiques

1. **Optimiser les prompts** : Éviter les répétitions et les phrases trop longues.
2. **Compresser les données** : Supprimer les espaces ou mots inutiles pour réduire le nombre de tokens.
3. **Choisir le bon modèle** :
 - **Llama 3.2** : Équilibre performance/coût.
 - **GPT-4** : Pour des contextes très longs (mais plus cher).
 - **Mistral** : Efficace pour le français et les tâches courtes.

6. Exemple concret : tokenisation d'un paragraphe

Texte : « Les tokens sont essentiels pour comprendre comment les modèles de langage comme Llama 3.2, Mistral ou GPT-4 traitent le texte. Chaque modèle a sa propre méthode de tokenisation, ce qui influence le nombre de tokens générés et donc le coût de traitement. »

Modèle	Nombre de tokens	Coût estimé (pour 1 000 tokens)
Llama 3.2	42	0,0042 \$
Mistral	39	0,00585 \$
GPT-4	45	0,0135 \$

▢ Points clés à retenir

1. **1 token ≠ 1 mot** : Tout dépend du modèle et de la langue.
2. **1 Mo ≈ 250 000 tokens** : Utile pour estimer les coûts et les limites.
3. **Choix du modèle** : À adapter selon la taille du texte, le budget et la langue.

La tokenisation est le pont invisible entre le langage humain et la compréhension machine. Bien la maîtriser, c’est optimiser chaque interaction avec un LLM.

▢ Sources

- [Attention Is All You Need](#) – Vaswani et al. (2017)
- [Hugging Face](#) – Tokenizers
- [OpenAI](#) – Tokenization Guide
- [Llama 3.2 Technical Report](#) – Meta AI (2024)
- [Mistral 7B Announcement](#) – Mistral AI (2023)

From:

<https://wiki.bozocinformatique.fr/> - Hack the knowledge, secure the future

Permanent link:

<https://wiki.bozocinformatique.fr/ia:tokenisation>

Last update:

2025/12/04 10:24

