Et si les Tokens étaient une Erreur de Conception ?

Byte Latent Transformer : l'architecture qui pourrait tout changer (Meta AI, 2024)

Amir KELLOU SIDHOUM
DENEM Labs

D'après le paper de Meta AI

Décembre 2025

Le Paradoxe qui Dérange

Demandez à ChatGPT combien de "r" dans "strawberry".

Réponse attendue

3

(s-t-**r**-a-w-b-e-**r**-**r**-y)

Réponse fréquente des LLM

2

"Il y a 2 lettres r dans strawberry"

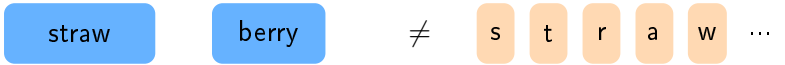
La question qui tue

Comment un modèle à 175 milliards de paramètres peut-il échouer sur une tâche qu'un enfant de 6 ans réussit ?

Le Coupable : La Tokenization

Ce que font TOUS les LLM actuels

Ils ne voient pas les caractères. Ils voient des **tokens**.



Pourquoi ? Compromis entre :

- Caractères seuls → trop de tokens, pas de sémantique
- Mots entiers → vocabulaire infini, typos impossibles
- Subwords (tokens) → le "juste milieu"... vraiment ?

L'Hypothèse que Personne ne Questionne

"La tokenization est nécessaire.
C'est le meilleur compromis possible."

L'argument classique

- Les tokens portent du sens sémantique
- Réduisent la longueur des séquences
- Permettent de gérer des vocabulaires raisonnables (32K-100K tokens)
- "Tout le monde fait ça" (GPT, Claude, Llama, Mistral...)

Et si c'était FAUX ?

Et si la tokenization était une béquille... pas une solution ?

Les Problèmes que la Tokenization Crée

1. **Incapacité à comprendre les caractères**
   - Compter des lettres → échec
   - Inverser un mot → difficile
   - Détecter des palindromes → aléatoire
2. **Gaspillage de compute massif**

Token simple	Token complexe
"."	**"quantum"**
1 passage transformer complet	1 passage transformer complet

Même coût. Même compute. C'est absurde.

Les Problèmes Cachés (suite)

3. Catastrophe multilingue

Anglais (sur-représenté)

"Hello world" → **2 tokens**

Chinois/Arabe (sous-représenté)

Même sens → **5-10 tokens**

→ Plus de tokens = plus cher = moins performant

4. Fragilité aux variations

"hello" → 1 token

"helo" → 3 tokens différents

Conséquence

Une simple typo peut complètement déstabiliser le modèle... et faciliter les jailbreaks.

Et si on supprimait complètement les tokens ?

BLT = Byte Latent Transformer

- Architecture **sans tokenizer**
- Travaille directement sur les **bytes bruts**
- Utilise des **patches dynamiques** au lieu de tokens fixes

LLM classique

Texte → Tokenizer → Tokens → Transformer

BLT

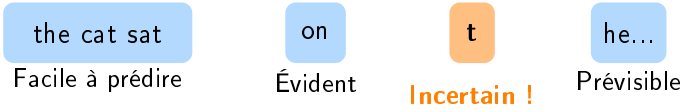
Texte → **Bytes** → **Patches** → Transformer

L'Idée Géniale : L'Entropie comme Guide

Le principe

Allouer plus de compute là où les **décisions sont difficiles**.

Exemple : "the cat sat on the"



Entropy-based patching

Les frontières des patches sont placées aux points de **haute incertitude** (entropie élevée) = là où le modèle hésite.

Analogie : Lire un Livre

LLM classique (tokens)

Lire à vitesse constante

Chaque mot = même temps
"Le" = "anticonstitutionnellement"
Inefficace

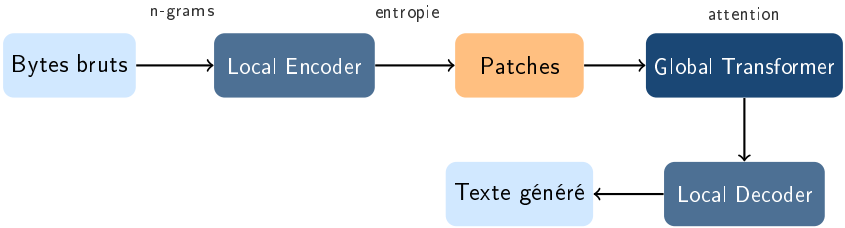
BLT (patches dynamiques)

Lire comme un humain

Passages simples → vite
Passages complexes → lentement
Naturel et efficace

C'est comme si vous pouviez **ajuster votre attention** en fonction de la difficulté du contenu, au lieu de traiter chaque mot de manière identique.

Architecture BLT : Vue d'Ensemble



Les 3 composants clés

1. **Local Encoder** : Transforme les bytes en patches (léger)
2. **Global Transformer** : Le "cerveau" - prédit les représentations
3. **Local Decoder** : Reconvertit en bytes (léger)

Les Résultats qui Font Mal

Performance

BLT 8B = Llama 3 8B

Même niveau de performance
sur les benchmarks standards

Efficacité

Jusqu'à 50% de FLOPs en moins

À inférence égale,
BLT consomme moitié moins

Chiffres du paper (Meta AI)

- **8 milliards** de paramètres
- **4 trillions** de bytes d'entraînement
- Premier modèle byte-level à cette échelle

Là où BLT Surpasse les LLM Classiques

Tâches "sub-token" (enfin résolues)

- **Orthographe** : comprend les caractères individuels
- **Phonologie** : perçoit les sons, pas juste les mots
- **Traduction low-resource** : langues sous-représentées

LLM tokenisé

"strawberry" = 1-2 tokens
→ Aucune visibilité sur les lettres

BLT

"strawberry" = bytes
→ Chaque caractère est accessible

Le bonus inattendu

Nouveau **axe de scaling** : on peut ajuster la taille des patches, pas seulement la taille du modèle !

Ce que ça Change pour l'IA

Avant BLT :

- Tokenizer = étape séparée, figée, entraînée à part
- Performance multilingue = dépend du corpus d'entraînement du tokenizer
- Comprendre les caractères = impossible by design

Avec BLT :

- Plus de tokenizer = architecture end-to-end
- Toutes les langues traitées équitablement
- Compréhension native du niveau caractère

La vraie révolution

Ce n'est plus "comment améliorer les tokens ?"
C'est "**a-t-on vraiment besoin de tokens ?**"

Les Questions qui Restent Ouvertes

Ce que BLT ne résout pas (encore)

- **Scaling** : Testé jusqu'à 8B, quid de 70B+ ?
- **Entraînement** : Plus coûteux que les modèles tokenisés ?
- **Adoption** : Tout l'écosystème est construit sur les tokens

Recherche récente (mai 2025)

D'autres approches émergent :

- Modèles qui prédisent **plusieurs bytes d'un coup**
- Segmentation par espaces (limité : ne fonctionne pas pour le chinois)

Le paradigme byte-level est en pleine effervescence.

Pour Vous, Concrètement

Si vous utilisez des LLM

- Les tâches de manipulation de caractères restent problématiques
- Vérifiez toujours les outputs sur du comptage/orthographe
- Les langues non-anglaises coûtent plus cher en tokens

Si vous construisez des produits IA

- Surveillez les modèles byte-level (BLT, successeurs)
- Le coût d'inférence pourrait baisser de 50%
- L'équité multilingue pourrait devenir un avantage compétitif

Le signal à retenir

Meta a open-sourcé BLT 8B. Le paradigme post-token commence.

Pour Aller Plus Loin

Paper original

"Byte Latent Transformer: Patches Scale Better Than Tokens"
Meta AI, 2024
`https://arxiv.org/abs/2412.09871`

Modèle open-source

BLT 8B disponible sur Hugging Face
`https://huggingface.co/meta-llama/blt`

À suivre

- L'équipe Meta FAIR (auteurs du paper)
- Recherches sur les architectures byte-level
- Évolutions de Llama intégrant ces concepts

La Question n'est Plus...

"Comment améliorer la tokenization ?"

Mais...

"Les tokens étaient-ils une erreur
depuis le début ?"

50% de compute en moins. Performance égale.
La réponse de Meta est claire.

Et vous ?

Vous pensez que les tokens vont disparaître ?

Ou que c'est "juste" une optimisation de plus ?

Dites-moi en commentaire.

Amir KELLOU-SIDHOUM | DENEM Labs

Souveraineté IA • Automatisation • LLM