LLM-Series

Tokenization

Tokenization là quá trình chia nhỏ văn bản gốc thành các phần nhỏ hơn gọi là **token**.
Các token này được ánh xạ thành ID số nguyên bằng một bộ từ vựng cố định.
Đây là **bước đầu tiên** trong mọi LLM.

Ví dụ

"unbelievable transformers"

từ hiếm bị chia thành các subword

tokenize

khoảng trắng là một phần của token

un	believable	transform	ers
4302	12871	8932	7421

token vs ký tự vs từ

a **ký tự:** quá ngắn, không có ngữ nghĩa

T **từ:** quá nhiều, không xử lý được từ mới

✦ **token:** điểm cân bằng - subword ⟵ giải pháp thực tế

★ mỗi mô hình có bộ tokenizer riêng;
kích thước từ vựng thường từ 30k-200k

Văn bản ⟶ số nguyên mà mô hình có thể sử dụng

LLM-Series

Token vs Từ vs Ký Tự

LLM không hiểu văn bản như con người.
Chúng chuyển văn bản thành **token** bằng một bộ tokenizer.
Token, từ, và ký tự **không** giống nhau!

TOKEN

✦

Các mảnh subword được tạo ra bởi tokenizer. Có thể là một từ đầy đủ hoặc một phần của từ.

Ví dụ

unbelievable transformers

↓

un | believable | transform | ers

→ 4 token

Dùng trong

LLM & Transformer
(GPT, Claude, Llama, v.v.)

TỪ

T

Một chuỗi ký tự cách nhau bởi khoảng trắng. Khớp với cách con người đọc văn bản.

Ví dụ

unbelievable transformers

↓

unbelievable | transformers

→ 2 từ

Dùng trong

Công cụ tìm kiếm, SEO,
NLP truyền thống

KÝ TỰ

A

Đơn vị nhỏ nhất của văn bản. Mỗi chữ cái, số, khoảng trắng, hoặc ký hiệu.

Ví dụ

unbelievable transformers

↓

u | n | b | e | l | i | ...

→ 22+ ký tự

Dùng trong

Kiểm tra chính tả, Regex,
Một số mô hình cổ điển

- ✓ Token là sự cân bằng giữa từ và ký tự.
- ✓ Chúng giúp mô hình xử lý từ hiếm và từ mới.
- ✓ Kích thước từ vựng của hầu hết LLM hiện đại: **30k - 200k token**

Token chính là ngôn ngữ mà LLM thực sự sử dụng.

Văn bản ⟶ tokenize ⟶ số nguyên ⟶ mô hình hiểu

LLM-Series

BPE
(Mã Hóa Cặp Byte)

BPE là một thuật toán tokenization xây dựng bộ từ vựng bằng cách lặp đi lặp lại việc **gộp các cặp byte** (hoặc ký tự) xuất hiện thường xuyên nhất. Đây là thuật toán đơn giản, hiệu quả, và được sử dụng rộng rãi trong các LLM hiện đại như GPT, Llama, Claude, và nhiều mô hình khác.

CÁCH BPE HOẠT ĐỘNG (Ví dụ)

Hãy tokenize từ: l o w e s t

	Cặp Xuất Hiện Nhiều Nhất	Gộp → Token Mới
1 Bắt đầu với các ký tự	l o w e s t Cặp: (l,o) (o,w) (w,e) (e,s) (s,t) 1 1 1 1 1	Tất cả các cặp xuất hiện 1 lần. (o,w) → **ow**
2 Gộp cặp xuất hiện nhiều nhất	l **ow** e s t Cặp: (l,ow) (ow,e) (e,s) (s,t) 1 1 1 1	(ow,e) → **owe**
3 Tiếp tục gộp	l **owe** s t Cặp: (l,owe) (owe,s) (s,t) 1 1 1	(l,owe) → **lowe**
4 Tiếp tục gộp	**lowe** s t Cặp: (lowe,s) (s,t) 1 1	(lowe,s) → **lowest**
5 Gộp cuối cùng	**lowest** t Cặp: (lowest,t) 1	(lowest,t) → **lowestt**

★ Token cuối cùng: "lowestt" (sẽ là một token duy nhất trong bộ từ vựng)
Trong các mô hình thực tế, ta dừng lại khi đạt **giới hạn kích thước từ vựng** (ví dụ: **50k, 100k**)

TẠI SAO DÙNG BPE?

- ✓ Xử lý được mọi từ (kể cả từ chưa từng thấy)
- ✓ Giảm kích thước từ vựng
- ✓ Cân bằng giữa ký tự và từ hoàn chỉnh
- ✓ Nhanh và có khả năng mở rộng

ĐƯỢC SỬ DỤNG Ở ĐÂU?

- GPT (tiktoken)
- Llama (SentencePiece - BPE)
- Claude
- Nhiều LLM hiện đại khác

BPE xây dựng một bộ từ vựng thông minh bằng cách học xem những đoạn văn bản nào nên được giữ cùng nhau.

Nén tốt hơn, khái quát hóa tốt hơn!

LLM-Series

Token Đặc Biệt

Token đặc biệt là những token được dành riêng với ý nghĩa cụ thể.
Chúng **không** phải là một phần của văn bản thông thường.
LLM sử dụng chúng để hiểu cấu trúc, kiểm soát hành vi, và xử lý các tác vụ khác nhau.

CÁC TOKEN ĐẶC BIỆT PHỔ BIẾN

Token	Ý nghĩa	Ví dụ / Cách dùng	Minh họa
<PAD>	Token đệm. Dùng để làm cho các chuỗi có cùng độ dài.	[1, 205, 89, <PAD>, <PAD>] (thêm vào cuối)	■□□
<BOS>	Bắt đầu chuỗi. Đánh dấu điểm bắt đầu của đầu vào.	<BOS> Xin chào, bạn khỏe không?	•→
<EOS>	Kết thúc chuỗi. Đánh dấu điểm kết thúc của đầu ra.	Tôi khỏe, cảm ơn! <EOS>	•→
<UNK>	Token không xác định. Dùng cho các từ chưa từng gặp.	blorple → <UNK>	?
<SEP>	Token phân tách. Phân tách hai phần.	Câu hỏi <SEP> Câu trả lời	□⁞□
<CLS>	Token phân loại. Đại diện cho toàn bộ đầu vào.	[<CLS> Bộ phim thật tuyệt!]	✦
<MASK>	Token che. Dùng trong mô hình ngôn ngữ có che từ (masked LM).	Tôi yêu <MASK>!	▣

TẠI SAO CẦN TOKEN ĐẶC BIỆT?

- ✓ Thêm cấu trúc và ranh giới
- ✓ Giúp mô hình hiểu ngữ cảnh
- ✓ Hỗ trợ nhiều tác vụ khác nhau (chat, dịch, phân loại, v.v.)
- ✓ Cải thiện quá trình huấn luyện và sinh văn bản

ĐƯỢC SỬ DỤNG Ở ĐÂU?

- Mô hình chat (system, user, assistant)
- Mô hình dịch (nguồn <SEP> đích)
- Phân loại chuỗi (<CLS>)
- Sinh văn bản (BOS, EOS)
- Mô hình ngôn ngữ có che từ (<MASK>)

Token đặc biệt chính là chất keo gắn kết hội thoại, cấu trúc, và trí tuệ của LLM lại với nhau.

Token nhỏ, tác động lớn!

LLM-Series

KÍCH THƯỚC TỪ VỰNG
SỰ ĐÁNH ĐỔI

Kích thước từ vựng là số lượng token duy nhất trong một tokenizer.
Chọn kích thước phù hợp là sự cân bằng giữa **hiệu quả**, **độ bao phủ**, **chi phí**, và **hiệu năng**.

Từ Vựng Nhỏ Hơn (~10k)	SỰ ĐÁNH ĐỔI	Từ Vựng Lớn Hơn (200k+)
Nhiều token hơn mỗi văn bản (chuỗi dài hơn)	**Độ Dài Chuỗi**	Ít token hơn mỗi văn bản (chuỗi ngắn hơn)
Chi phí tính toán cao hơn (nhiều token cần xử lý)	**Hiệu Quả & Tốc Độ**	Chi phí tính toán thấp hơn (huấn luyện & suy luận nhanh hơn)
Tốt hơn với từ hiếm, lỗi chính tả, và từ mới	**Độ Bao Phủ**	Tốt hơn với từ phổ biến và ngữ nghĩa nguyên từ
Có thể chia từ thành nhiều mảnh (kém mạch lạc)	**Tính Mạch Lạc Ngữ Nghĩa**	Nắm bắt nguyên từ / cụm từ (mạch lạc hơn)
Cửa sổ ngữ cảnh lớn hơn tính theo ký tự	**Cửa Sổ Ngữ Cảnh (Ký tự vs Token)**	Cửa sổ ngữ cảnh nhỏ hơn tính theo ký tự
Ma trận embedding nhỏ hơn (ít tham số hơn)	**Kích Thước Mô Hình (Embedding)**	Ma trận embedding lớn hơn (nhiều tham số hơn)

VÍ DỤ (Khoảng Điển Hình)

10k - 20k
- Cấp độ ký tự/từ
- Rất tốt cho từ hiếm
- Nhiều token hơn
- Chậm hơn & tốn kém hơn

30k - 50k
- Lựa chọn cân bằng
- Độ bao phủ tốt
- Hiệu quả
- Khoảng phổ biến nhất

100k - 150k
- Ít token hơn
- Ngữ nghĩa tốt hơn
- Nhanh hơn & rẻ hơn
- Kích thước embedding lớn

200k+
- Rất ít token
- Tốt nhất cho hiệu quả
- Kích thước embedding khổng lồ
- Lợi ích giảm dần

ĐIỂM CHÍNH CẦN NHỚ

✓ Không có kích thước phù hợp cho tất cả.
✓ Phụ thuộc vào mô hình, dữ liệu, và trường hợp sử dụng.
✓ Từ vựng nhỏ → linh hoạt hơn, nhiều token hơn.
✓ Từ vựng lớn → hiệu quả hơn, giàu ngữ nghĩa hơn.
✓ Hầu hết LLM hiện đại dùng 30k - 200k token.

ĐIỀU GÌ QUAN TRỌNG NHẤT?

✓ Chất lượng mô hình > chỉ đơn thuần từ vựng lớn
✓ Thiết kế tokenizer tốt > kích thước từ vựng thô
✓ Cân bằng: hiệu quả, độ bao phủ, và chi phí
✓ Chọn cái phù hợp nhất với trường hợp sử dụng của bạn!

Mục tiêu không phải là **từ vựng lớn nhất**, mà là **từ vựng phù hợp nhất** cho công việc.

LLM-Series

Tokenization

Tokenization là quá trình chia nhỏ văn bản gốc thành các phần nhỏ hơn gọi là **token**.
Các token này được ánh xạ thành ID số nguyên bằng một bộ từ vựng cố định.
Đây là **bước đầu tiên** trong mọi LLM.

Ví dụ

"unbelievable transformers"

từ hiếm bị chia thành các subword

tokenize

khoảng trắng là một phần của token

un	believable	transform	ers
4302	12871	8932	7421

token vs ký tự vs từ

a **ký tự:** quá ngắn, không có ngữ nghĩa

T **từ:** quá nhiều, không xử lý được từ mới

✦ **token:** điểm cân bằng - subword ⟵ giải pháp thực tế

★ mỗi mô hình có bộ tokenizer riêng;
kích thước từ vựng thường từ 30k-200k

Văn bản ⟶ số nguyên mà mô hình có thể sử dụng

LLM-Series

Token vs Từ vs Ký Tự

LLM không hiểu văn bản như con người.
Chúng chuyển văn bản thành **token** bằng một bộ tokenizer.
Token, từ, và ký tự **không** giống nhau!

TOKEN

✦

Các mảnh subword được tạo ra bởi tokenizer. Có thể là một từ đầy đủ hoặc một phần của từ.

Ví dụ

unbelievable transformers

↓

un | believable | transform | ers

→ 4 token

Dùng trong

LLM & Transformer
(GPT, Claude, Llama, v.v.)

TỪ

T

Một chuỗi ký tự cách nhau bởi khoảng trắng. Khớp với cách con người đọc văn bản.

Ví dụ

unbelievable transformers

↓

unbelievable | transformers

→ 2 từ

Dùng trong

Công cụ tìm kiếm, SEO,
NLP truyền thống

KÝ TỰ

A

Đơn vị nhỏ nhất của văn bản. Mỗi chữ cái, số, khoảng trắng, hoặc ký hiệu.

Ví dụ

unbelievable transformers

↓

u | n | b | e | l | i | ...

→ 22+ ký tự

Dùng trong

Kiểm tra chính tả, Regex,
Một số mô hình cổ điển

- ✓ Token là sự cân bằng giữa từ và ký tự.
- ✓ Chúng giúp mô hình xử lý từ hiếm và từ mới.
- ✓ Kích thước từ vựng của hầu hết LLM hiện đại: **30k - 200k token**

Token chính là ngôn ngữ mà LLM thực sự sử dụng.

Văn bản ⟶ tokenize ⟶ số nguyên ⟶ mô hình hiểu

LLM-Series

BPE
(Mã Hóa Cặp Byte)

BPE là một thuật toán tokenization xây dựng bộ từ vựng bằng cách lặp đi lặp lại việc **gộp các cặp byte** (hoặc ký tự) xuất hiện thường xuyên nhất. Đây là thuật toán đơn giản, hiệu quả, và được sử dụng rộng rãi trong các LLM hiện đại như GPT, Llama, Claude, và nhiều mô hình khác.

CÁCH BPE HOẠT ĐỘNG (Ví dụ)

Hãy tokenize từ: l o w e s t

	Cặp Xuất Hiện Nhiều Nhất	Gộp → Token Mới
1 Bắt đầu với các ký tự	l o w e s t Cặp: (l,o) (o,w) (w,e) (e,s) (s,t) 1 1 1 1 1	Tất cả các cặp xuất hiện 1 lần. (o,w) → **ow**
2 Gộp cặp xuất hiện nhiều nhất	l **ow** e s t Cặp: (l,ow) (ow,e) (e,s) (s,t) 1 1 1 1	(ow,e) → **owe**
3 Tiếp tục gộp	l **owe** s t Cặp: (l,owe) (owe,s) (s,t) 1 1 1	(l,owe) → **lowe**
4 Tiếp tục gộp	**lowe** s t Cặp: (lowe,s) (s,t) 1 1	(lowe,s) → **lowest**
5 Gộp cuối cùng	**lowest** t Cặp: (lowest,t) 1	(lowest,t) → **lowestt**

★ Token cuối cùng: "lowestt" (sẽ là một token duy nhất trong bộ từ vựng)
Trong các mô hình thực tế, ta dừng lại khi đạt **giới hạn kích thước từ vựng** (ví dụ: **50k, 100k**)

TẠI SAO DÙNG BPE?

- ✓ Xử lý được mọi từ (kể cả từ chưa từng thấy)
- ✓ Giảm kích thước từ vựng
- ✓ Cân bằng giữa ký tự và từ hoàn chỉnh
- ✓ Nhanh và có khả năng mở rộng

ĐƯỢC SỬ DỤNG Ở ĐÂU?

- GPT (tiktoken)
- Llama (SentencePiece - BPE)
- Claude
- Nhiều LLM hiện đại khác

BPE xây dựng một bộ từ vựng thông minh bằng cách học xem những đoạn văn bản nào nên được giữ cùng nhau.

Nén tốt hơn, khái quát hóa tốt hơn!

LLM-Series

Token Đặc Biệt

Token đặc biệt là những token được dành riêng với ý nghĩa cụ thể.
Chúng **không** phải là một phần của văn bản thông thường.
LLM sử dụng chúng để hiểu cấu trúc, kiểm soát hành vi, và xử lý các tác vụ khác nhau.

CÁC TOKEN ĐẶC BIỆT PHỔ BIẾN

Token	Ý nghĩa	Ví dụ / Cách dùng	Minh họa
<PAD>	Token đệm. Dùng để làm cho các chuỗi có cùng độ dài.	[1, 205, 89, <PAD>, <PAD>] (thêm vào cuối)	■□□
<BOS>	Bắt đầu chuỗi. Đánh dấu điểm bắt đầu của đầu vào.	<BOS> Xin chào, bạn khỏe không?	•→
<EOS>	Kết thúc chuỗi. Đánh dấu điểm kết thúc của đầu ra.	Tôi khỏe, cảm ơn! <EOS>	•→
<UNK>	Token không xác định. Dùng cho các từ chưa từng gặp.	blorple → <UNK>	?
<SEP>	Token phân tách. Phân tách hai phần.	Câu hỏi <SEP> Câu trả lời	□⁞□
<CLS>	Token phân loại. Đại diện cho toàn bộ đầu vào.	[<CLS> Bộ phim thật tuyệt!]	✦
<MASK>	Token che. Dùng trong mô hình ngôn ngữ có che từ (masked LM).	Tôi yêu <MASK>!	▣

TẠI SAO CẦN TOKEN ĐẶC BIỆT?

- ✓ Thêm cấu trúc và ranh giới
- ✓ Giúp mô hình hiểu ngữ cảnh
- ✓ Hỗ trợ nhiều tác vụ khác nhau (chat, dịch, phân loại, v.v.)
- ✓ Cải thiện quá trình huấn luyện và sinh văn bản

ĐƯỢC SỬ DỤNG Ở ĐÂU?

- Mô hình chat (system, user, assistant)
- Mô hình dịch (nguồn <SEP> đích)
- Phân loại chuỗi (<CLS>)
- Sinh văn bản (BOS, EOS)
- Mô hình ngôn ngữ có che từ (<MASK>)

Token đặc biệt chính là chất keo gắn kết hội thoại, cấu trúc, và trí tuệ của LLM lại với nhau.

Token nhỏ, tác động lớn!

LLM-Series

KÍCH THƯỚC TỪ VỰNG
SỰ ĐÁNH ĐỔI

Kích thước từ vựng là số lượng token duy nhất trong một tokenizer.
Chọn kích thước phù hợp là sự cân bằng giữa **hiệu quả**, **độ bao phủ**, **chi phí**, và **hiệu năng**.

Từ Vựng Nhỏ Hơn (~10k)	SỰ ĐÁNH ĐỔI	Từ Vựng Lớn Hơn (200k+)
Nhiều token hơn mỗi văn bản (chuỗi dài hơn)	**Độ Dài Chuỗi**	Ít token hơn mỗi văn bản (chuỗi ngắn hơn)
Chi phí tính toán cao hơn (nhiều token cần xử lý)	**Hiệu Quả & Tốc Độ**	Chi phí tính toán thấp hơn (huấn luyện & suy luận nhanh hơn)
Tốt hơn với từ hiếm, lỗi chính tả, và từ mới	**Độ Bao Phủ**	Tốt hơn với từ phổ biến và ngữ nghĩa nguyên từ
Có thể chia từ thành nhiều mảnh (kém mạch lạc)	**Tính Mạch Lạc Ngữ Nghĩa**	Nắm bắt nguyên từ / cụm từ (mạch lạc hơn)
Cửa sổ ngữ cảnh lớn hơn tính theo ký tự	**Cửa Sổ Ngữ Cảnh (Ký tự vs Token)**	Cửa sổ ngữ cảnh nhỏ hơn tính theo ký tự
Ma trận embedding nhỏ hơn (ít tham số hơn)	**Kích Thước Mô Hình (Embedding)**	Ma trận embedding lớn hơn (nhiều tham số hơn)

VÍ DỤ (Khoảng Điển Hình)

10k - 20k
- Cấp độ ký tự/từ
- Rất tốt cho từ hiếm
- Nhiều token hơn
- Chậm hơn & tốn kém hơn

30k - 50k
- Lựa chọn cân bằng
- Độ bao phủ tốt
- Hiệu quả
- Khoảng phổ biến nhất

100k - 150k
- Ít token hơn
- Ngữ nghĩa tốt hơn
- Nhanh hơn & rẻ hơn
- Kích thước embedding lớn

200k+
- Rất ít token
- Tốt nhất cho hiệu quả
- Kích thước embedding khổng lồ
- Lợi ích giảm dần

ĐIỂM CHÍNH CẦN NHỚ

✓ Không có kích thước phù hợp cho tất cả.
✓ Phụ thuộc vào mô hình, dữ liệu, và trường hợp sử dụng.
✓ Từ vựng nhỏ → linh hoạt hơn, nhiều token hơn.
✓ Từ vựng lớn → hiệu quả hơn, giàu ngữ nghĩa hơn.
✓ Hầu hết LLM hiện đại dùng 30k - 200k token.

ĐIỀU GÌ QUAN TRỌNG NHẤT?

✓ Chất lượng mô hình > chỉ đơn thuần từ vựng lớn
✓ Thiết kế tokenizer tốt > kích thước từ vựng thô
✓ Cân bằng: hiệu quả, độ bao phủ, và chi phí
✓ Chọn cái phù hợp nhất với trường hợp sử dụng của bạn!

Mục tiêu không phải là **từ vựng lớn nhất**, mà là **từ vựng phù hợp nhất** cho công việc.