

從構式語法看「token」

tokenizer 作為一種隱含的詞庫理論

Week 3 · 補充 B

這一節要討論的

一個 BPE 詞表其實不只是工程參數，是一份詞庫清單。既然是清單，就可以問它依什麼判準收錄、收了什麼、漏了什麼。

主張：

BPE 實作了 J. Bybee 的固化機制，卻沒有實作範疇化機制。它把高頻的具體字串變成單位，但永遠推不出構式。

這不是說 BPE 錯了，而是說它決定了哪些東西必須由模型在後續層重建，以及重建的代價。

一、先釐清「token」這個詞

三個不同的詞義

J&K 課本從 Peirce 的 type / token 區分講起，然後 NLP 把這個詞用壞了。

用法	指的是	應該叫
「詞表有 20 萬個 token」	詞表的條目	type / vocabulary item
「這句話有 13 個 token」	文本中的出現	token occurrence
「tokenization 的結果」	切分出來的片段	segment

明確區分, 否則「成語不是一個 token」這句話會同時有三種讀法，而三種讀法之後課堂上都會碰到。

一個更根本的差異要注意

Peirce 的 type 是由 token 的相似性歸納出來的範疇。

BPE 的 vocabulary item 是由字串頻率切出來的形式。

前者預設了一個歸類的判準（什麼算「同一個」）。後者沒有：他的 和 更新 在詞表裡地位完全相同，

雖然一個是詞、一個不是。

詞表是一份沒有範疇判準的 type 清單。

二、tokenizer 承諾了什麼

把 BPE 的假設

一個 BPE 詞表隱含地主張，語言的基本單位是：

1. 實體的 (substantive) — 有具體的字形材料，不是變數
2. 連續的 (contiguous) — 表層必須相鄰
3. 由頻率定義的 — 夠常見就是單位，不常見就不是
4. 與語境無關的 — 同一字串在哪裡都切成一樣
5. 不帶意義的 — 單位只有形式，沒有配對的語義

第五點是最大的斷裂

結構主義：語素 = 最小的表義單位（minimal meaning-bearing unit）。

BPE：單位 = 最小的表頻單位。

把 *meaning* 換成 *frequency*，是整個現代 NLP 最關鍵的一次替換。

它讓「找出單位」變得可計算，代價是單位不再需要有意義。

以前Week 2 講的「誰有權定義語言單位」，在這裡看出答案：由一個不看意義的計數程序定義。

三、構式語法 (C*G) 提供的座標

CxG 的三個主張

1. 詞庫與句法是一個連續體

Fillmore, Kay & O'Connor (1988) 的 *let alone* ; Goldberg (1995, 2006) ; Croft (2001) 。

所有層次的單位都是形式—意義配對，差別只在大小與抽象度。

2. 圖式性是一個維度

一石二鳥（全實體）… [V 得 A]（部分圖式）… [主 動 賓]（全圖式）

3. 固化(entrenchment)與能產性(productivity)由不同的頻率驅動 Bybee (1985, 2006)：

- token frequency（特定字串出現多少次）→ 驅動該字串的固化
- type frequency（同一圖式有多少不同填充）→ 驅動圖式化與能產性

關鍵點

BPE 只能看到連續字串的 token frequency。它沒有辦法看到 type frequency，因為那需要先知道

「越走越快、越說越氣、越想越不對」是同一個東西的不同填充」，而那正是它沒學的東西。

BPE 有的：固化機制 高頻字串 → 單位

BPE 沒有的：範疇化機制 多樣填充 → 圖式

把它畫成兩個維度

	連續	不連續
實體	一石二鳥、巧克力、他的	洗…澡、幫…忙（離合詞）
圖式	[V 得 A]、[A 不 A]	越 X 越 Y、一 X 就 Y、連 NP 都 V

BPE 能表徵的只有左上角，而且左上角裡還要夠高頻。

四、實證證據

證據 1：V-not-V 的固化梯度

o200k_base 的 token 數：

是不是	1	好不好	2	要不要	2	能不能	2
對不對	3	想不想	3	去不去	3	來不來	3

同一個構式 [A 不 A]，八個實例，四種待遇。

cl100k_base 的同一組：

是不是	3	好不好	3	要不要	3	能不能	2
對不對	5	想不想	5	去不去	3	來不來	5

注意排序變了：**能不能** 反而最低。因為在 cl100k 裡，差異主要由「這幾個字有沒有自己的 token」決定，頻率梯度被字形覆蓋率蓋過去了。

兩個機制疊在一起：覆蓋率決定地板，固化決定誰能往上爬。兩個詞表裡，圖式都沒有被學到。只有最高頻的實例被收進詞表。

證據 2：重疊

o200k_base：

看看 1 等等 1
聽聽 4

想想 2 說說 2 試試 2 走走 2 問問 2
研究研究 2 (=研究+研究)

動詞重疊在漢語裡高度能產：幾乎任何單音節動作動詞都可以重疊。

但詞表只收了 看看 與 等等 一而 等等 早已詞彙化成「等等、諸如此類」，未必還算重疊式。真正作為重疊實例被收進去的，大概只有 看看。

聽聽 要 4 個 token，因為 聽 自己在詞表裡就沒有位置。

構式的待遇又被字形覆蓋率再乘一次。

研究研究 被切成 研究 + 研究：重疊的結果被還原成兩個底層形式，重疊這件事本身在表徵上消失了。

證據 3：成語

o200k_base :

一石二鳥 4	畫蛇添足 4	亡羊補牢 4	莫名其妙 4	一言難盡 5
無論如何 3	總而言之 3	(部分合併)		

沒有任何一個四字成語是單一 token。

成語是漢語裡固化程度最高的單位 — 語義不透明、形式換不得、心理上是整塊取出來的。而詞表一個都沒收。

原因很單純：`一石二鳥` 這個字串的頻率，遠低於 `最近`、`更新`、`他的` 這種 chunk。固化程度與字串頻率是兩回事，而 BPE 只看得到後者。

證據 4：merge rank 就是頻率序

BPE 的 merge 編號是學習順序，也就是頻率順序。o200k_base：

中文	10667	可以	15578	如何	24608	更新	26889
研究	39315	最近	65825	一起	94420	如此	94932
看看	110218	是不是	121107	沒有	177582	他的	180222

他的 排在 180222，勉強擠進 20 萬。它不是一個詞。

沒有（繁體）排 177582，而簡體 没有 早得多。

這份清單沒有任何一個判斷是語言學的。

它是「這個字串在某個爬蟲抓到的網頁裡出現了幾次」的排序。

證據 5：不連續性

洗澡	→	洗	澡	洗了個澡	→	洗	了	個	澡
生氣	→	生	氣	生他的氣	→	生	他的	氣	

生他的氣 這一行值得拉出來看：

tokenizer 把插入成分（他的）認成一個單位，
卻把構式的固定錨點（生...氣）切開。

越來越 三個 token，越走越快 四個——
構式的兩個 越 從來沒有被關聯起來。

這不是 bug，而是「以表層連續性當單位判準」必然會得到的結果。

五、反方論證

也許 tokenization 根本不需要表徵構式

一個誠實的反駁：

誰說單位必須在第 0 層就存在？

Transformer 有幾十層，注意力可以跨距離。

構式可以在模型內部被重建。

而且我們自己的資料支持這個反駁。

Yeh, Wang & Hsieh (2026)

用 sparse autoencoder 分析非字面語言在各層的處理，觀察到一個詞彙熟悉度梯度：

- 成語：第一層就分化
- 隱喻、語義變遷俚語：早期層的窄帶重新分析
- 新詞：需要在深層建構，表徵特徵大約是三倍

成語沒有進詞表，卻在第一層就被辨識出來。那個固化是模型學出來的，不是詞表給的。

所以論點要修正

不是「BPE 錯了」，而是：

Tokenization 設定了構式學習的初始條件。

它決定哪些單位是免費的、哪些必須被重建、以及重建要花幾層與多少特徵。

進了詞表 = 固化被硬編碼在第 0 層。

沒進詞表 = 固化必須在後續層重建，成本是深度與表徵資源。

這個版本比較弱，但可以驗證，而且與已有的機制詮釋研究對得上。

於是我們可以問

如果成語不在詞表卻在第一層分化，那：

1. 詞表有收的多字單位（是不是、看看），在第幾層分化？更早嗎？
2. 同一個構式的高頻實例（是不是）與低頻實例（想想），處理深度是否不同？如果是，那是詞表造成的還是語料頻率造成的？
3. 把同一個模型換成不同 tokenizer 續訓，構式的處理深度會不會移動？

第 2 題可以用現成模型做，第 3 題需要算力，但實驗設計很清楚。

六、往前看

一個「構式 tokenizer」需要什麼

1. 槽位 — 單位必須能是 越 __ 越 __ ，而不只是字串
2. 上下文敏感 — 洗澡 在 洗了個澡 裡是不是單位，要看語境
3. 對不連續敏感 — 錨點可以隔著材料
4. 型頻率——需要看到「同一個圖式的不同填充」

前三項技術上有部分答案，第四項沒有。

多詞單位（MWE）從 Sag et al. (2002) ⚠ 那篇

「a pain in the neck for NLP」之後二十多年，仍然是痛處。

動態切分正在往這個方向走（一半）

BLT、H-Net 這類方法把切分從型層次搬到符層次：
同一個字串在不同語境會有不同的切法。

那正是第 2 項。

但它們切的仍然是連續的 patch。

第 1 項（槽位）與第 3 項（不連續）還是沒解。

最前線的技術工作，在沒讀過構式語法的情況下，自己走到了構式語法要求的一半性質。

構式庫是有語域的

詞表不是「漢語的構式庫」，是某個語料在某個時間的構式庫。

台灣 = 1 token (rank 75626)
重慶 = 1 token (rank 6554)
廣州 = 1 token (rank 132426)
北京 = 1 token (rank 19340)

台灣 / 臺灣 = 2 tokens
重慶 = 3 tokens
廣州 = 2 tokens
高雄 · 新竹 · 台北 = 2 tokens

而 o200k_base 裡最長的中文 token 是
微信公众号天天中彩票 、 无码一区二区三区 。

這份構式庫屬於某個特定的網路語域。它現在成了全世界用 GPT 的人的預設詞庫。

接回 Week 2

誰有權定義語言單位？

時期	權威	判準
1898	馬建忠 《馬氏文通》	從拉丁語法借來的詞類
1968	趙元任	語法行為、可分離性
1990s	CKIP / PKU / MSRA / CTB	標註規範的可操作定義
2016–	BPE	語料中的字串頻率
2024–	熵切分 / 學習式 chunking	模型自身的可預測性

每一次，判準都離語言學論證更遠，離可計算的代理量更近。而每一次，新的判準都比舊的更「有效」。這是這門課要處理的語言學理論與計算之間的張力。

可做的題目

H1 固化 vs 頻率：四字成語與頻率配對的非固化四字串，token 數有沒有差別？（預測：沒有。固化程度對 BPE 不可見。）

H2 圖式天花板：半圖式構式的固定錨點是否從不合併？跨 tokenizer 家族是否一致？

H3 梯度與能產性：[A 不 A] 各實例的 token 數，與語料中該實例的頻率、以及該構式的能產性指標，分別相關多少？

H4 詞表的單位類型分布：抽樣 200 個多字漢字 token，兩人標註為 語素／詞／搭配／短語／碎片／套版文字，報告 κ 與分布。

H5 深度：詞表內與詞表外的多字單位，處理深度是否不同？（接 Yeh et al. 2026）

討論題

1. 如果「單位」只是某個計數程序的產物，那語言學的「詞」還剩下什麼工作？
2. 動態切分讓單位變成上下文相關的。這算是往構式語法靠近，還是只是把同一個頻率邏輯搬到推論時？
3. 一個不帶意義的形式清單，可以算是 construction 嗎？
如果不算，那模型在哪一層才開始有 construction？

讀什麼（請自行查核書目）

構式語法：Fillmore, Kay & O'Connor 1988 · Goldberg 1995, 2006 · Croft 2001
· Bybee 1985, 2006（固化與能產性）

漢語構式與單位：趙元任 1968 · 離合詞與 V-not-V 的相關文獻
· Week 2 的斷詞標準材料

多詞單位：Sag et al. 2002 ⚠

接著：Yeh, Wang & Hsieh 2026 ⚠（詞彙熟悉度與處理深度）

下週

從「什麼是單位」到「單位怎麼變成向量」

本週作業

- 重點預習 J&K Chapter 4-5
- 兩週內完成 A1 作業

老師的研究補充

單位該怎麼被學出來

這幾頁不是本週的教學重點。前面診斷了 BPE 缺範疇化、缺槽位；這裡只是把缺的東西寫成一個研究構想。

問題從哪裡來

構式語法裡，人學的是配對：

⟨FORM, MEANING/FUNCTION⟩

LM 預訓練要的是：

$$\min_{\theta} - \sum_t \log P_{\theta}(x_t \mid x_{<t})$$

不是「LM 沒辦法學意義」。是我們通常沒有把 meaning 當成獨立的監督訊號，只讓模型從形式預測裡自己誘導出潛在意義。

先分清三個問題

	問的是
1	next-token 會不會逼出功能上等價於構式的配對？
2	那個 meaning 有沒有 grounding？
3	單位為什麼要在學意義之前先切好？

第 1 題：可能 emerge，且分散在 residual / heads / MLP。

第 2 題：純文字沒有；人還有知覺、行動、意圖。

第 3 題更根本：單位本身該不該被預先決定。

一個常被講錯的流程

✗ : tokenization → 學好 embedding → 再訓練 LLM。

👍 : 先學 tokenizer，embedding 隨機初始化，再跟 Transformer 一起靠 next-token 更新。

Tokenizer 在任何「理解」開始之前，已經替模型決定了一套 form ontology。

unbelievable → un|believ|able，吃了飯 → 吃|了|飯：

這是形式切分，不保證是語言單位，更不是構式。

跟 usage-based 學習的對照

LLM：先切形式，再學 token 的分布 construction 若出現，是後來長出來的

**人：(usage-based) **utterance + 情境 + 意圖 形式與意義一起切出來 單位發現 $\Leftrightarrow$ 意義發現

單位的發現與意義的發現，是互相制約的。

尖銳的問題因此不是「為什麼不能同時學 form 和 meaning」，而是：

Why is tokenization independent of semantic / constructional learning?

不要再叫它 tokenizer

問題從「字串該在哪裡切開」，變成「哪一段反覆出現的形式，因為帶著反覆出現的功能，才配稱為單位」。

比較準確的名字：Construction Inducer（構式誘導器）。

	在問什麼
BPE	$u^* = \arg \max$ 頻率／壓縮
構式	$u^* = \arg \max$ 形式連貫 + 意義連貫 + 預測效用

再加一項： $C = \langle F, M, \mathcal{D} \rangle$

$\mathcal{D}$ = 槽位能換什麼、能產性、與其他構式的關係。

第一層其實不必太激進

底層仍可保留 byte / subword。kick the bucket 不必指望深層自己組裝成 DIE；
可以在上面長出第二層：

bytes → subwords → constructions

真正要跳的一步：單位可以有變量。

[by the way]	全實體
[the X-er, the Y-er]	半圖式
[NP V NP NP]	全圖式

Token 第一次被允許帶變數，這是整套構想最重要的一步。

輸出不該是一條序列

同一句 `John gave Mary a book` 可以同時是：

詞項 · AGENT / RECIPIENT / THEME ·

`[NP GIVE NP NP]` · `[NP V NP NP]`

Representation 是 **construction lattice**，不是 **sequence**。一刀切開的 tokenizer 強迫只留一種粒度；構式語法要的是連續體上的多層分析。

怎麼學：不要把 Goldberg 理論焊進去

人工塞構式 = 把語言理論 hard-code 進模型。有理論價值的是 usage-based induction。

觀察 (x, z) ， z = 情境／意義／世界狀態。找反覆出現的 c ，使 $I(c; z)$ 夠高；再用壓縮管住爆發的 pattern：

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda_1 L(\mathcal{G}) + \lambda_2 L(D \mid \mathcal{G}) - \lambda_3 I(C; M)$$

構式語法 + usage-based + 最短描述長度。

Meaning 從哪裡來，才是關鍵的一關

純文字： $M \approx$ 分布語意狀態，那只是比 BPE 更懂一點語意的切分。

若是 multimodal / agent，meaning 可以來自世界狀態的改變：

Give me the red block.

$$S_t \xrightarrow{\text{utterance}} A_t \rightarrow S_{t+1}$$

$\Rightarrow$ [give X Y] $\leftrightarrow$ TRANSFER

不是人工貼上的 semantic label，是從狀態轉移長出來的。

一句尚待驗證的話

Prediction learns linguistic regularities;
interaction learns constructions.

實驗草圖：兩組小模型看同一批語言資料。一組只有 next-token；另一組還必須依 utterance 行動，並得到環境回饋。看誰形成更穩的構式表徵、能否做 novel verb generalization、會不會出現 construction coercion。

這句目前是假設，不是事實。但值得做成實驗。

所以它不是詞表，是會生長的構式庫

$$\mathcal{G}_C = (C, R)$$

C = 構式， R = 繼承／相似／特化

新經驗可以長出新構式；兩個可以合併；既有構式可以更圖式化，也可以特化。

研究定位不要說小了。不是「用構式語法改進 tokenizer」，而是：

Learn what counts as a unit by learning what it means.

挑戰的是 tokenize first, learn meaning later 這條假設。

要先畫清的界線

若沒先跟這些工作分開，reviewer 會說早就有人做過：

- semantic tokenization · multi-token prediction
- latent / adaptive tokenization · neural chunking
- 補充 A 的 BLT、H-Net（動態，但仍是連續 patch）

差別在於：單位是否必須是形式—功能配對、是否允許槽位與不連續、以及 meaning 是否可以來自世界狀態的改變。

備註名稱：Construction Learning for Language Agents.