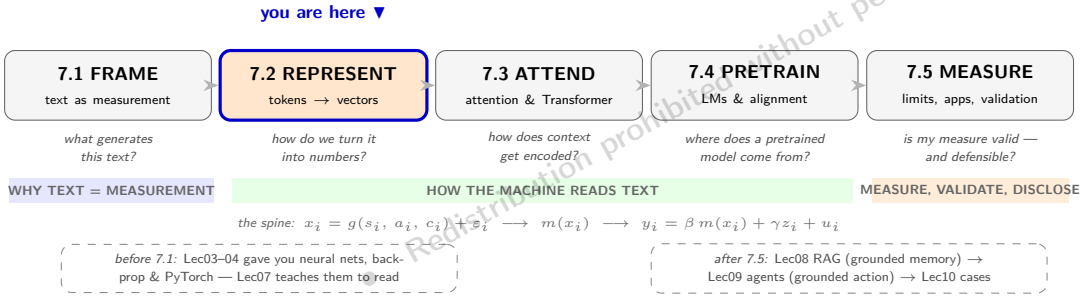
AI for Economic Research: Dynamic Models, Language, and Agents
Lecture 7.2a: Preprocessing and Bag-of-Words

Zhigang Feng

2026-07-09 05:17:58

Lecture 7 in One Map



Route: raw strings → clean text → tokens → counts (part 1/2 of REPRESENT — this deck builds the transparent count-based representations; part 2/2 turns counts into dense vectors).

From Strings to Tokens

Tokenization, subwords — and the Chinese special case

Tokenization: The First Step

- **Goal:** convert raw character sequences into discrete *tokens* that a neural network can consume.
- **Why not just split by spaces?**
  - Many economically relevant terms combine words, numbers, and symbols (e.g. `S&P500`, `CO2`, `Q4_2025`, `stagflation`)
  - Punctuation and formatting carry meaning ('$1.2B' vs '1.2 billion')
  - Different languages have different word boundaries (or none at all)
- **What is a token?** A flexible unit:
  - A whole word ('inflation')
  - A subword ('infl', 'ation')
  - A single character or byte
  - A special marker (`<EOS>`, `<PAD>`, `<USER>`)

Vocabulary and the Tokenization Map

- Let A be the alphabet (Unicode characters). A tokenizer is a function:

$$\mathcal{K}\colon A^* \longrightarrow \{1, 2, \ldots, M\}^*$$

  mapping any text string into a sequence of **numbers** — each number is a positive integer that serves as an index into a lookup table of size M. The fact that the index is an integer is incidental; what matters is that it *uniquely identifies* each token, the same way a ZIP code uniquely identifies a location.

- **Why integers?** Neural networks need numeric inputs; each token ID will index into an embedding lookup table.
- **Practical constraint — context length:**
  - Models have a maximum input size n (typically $2{,}048$ to $128{,}000$ tokens; some 1M)
  - Long policy documents, FOMC transcripts, or 10-K filings often exceed n
  - Forces *chunking*, *summarization*, or *retrieval* (a hint at Lec08)
- Vocabulary sizes are typically $30{,}000$–$100{,}000$ subword tokens. **To put this in perspective:** Merriam-Webster Unabridged lists $\approx 470{,}000$ entries; a college-educated native speaker actively uses roughly 20,000–40,000 words. A subword vocabulary of $M = 50{,}000$ therefore spans roughly

Illustrative Example: "GDP rose by 2%"

- A simple tokenization might produce

  $$(\text{GDP},\ \text{rose},\ \text{by},\ 2,\ \%)$$

  mapping to integer IDs, e.g. $(541,\ 823,\ 71,\ 312,\ 49)$.

- **Why integer mapping matters:**
  - Each ID becomes the index of a learned embedding vector
  - Numeric tokens ('2', '3', '5') get *their own* embeddings

- **Injectivity matters for forecasting:** a well-designed tokenizer maps
  - "inflation rate is 3.5%"
  - "inflation rate is 3.6%"

  to *different* token sequences. Otherwise small but economically meaningful differences would collapse into the same input.

- **Tokenizer/model coupling:** the tokenizer is part of the model. If you swap the tokenizer, all the embedding rows are meaningless — this is why every published LLM ships with its tokenizer.

Chinese Word Segmentation: Boundaries Must Be Inferred

English gets word boundaries for free; Chinese does not — so segmentation is itself a model.

- Chinese runs characters together; word boundaries must be *inferred*. Segmentation is a sequence-labeling problem: character by character, where does a word end?
- The classic ambiguity — 研究生命科学 splits as
  - 研究 | 生命科学 — "research *on* life science", or
  - 研究生 | 命科学 — "graduate student" + a nonsense remainder.

  Economic text has its own: 行业政策支持 parses as 行业政策 | 支持 ("industry-policy" + "support") or 行业 | 政策支持 ("industry" + "policy-support") — different head word, different meaning.
- **Tools:** Jieba, THULAC, LTP — each a dictionary plus a statistical/neural sequence labeler, trading speed against accuracy.
- **The economist's trap — domain compounds:** terms like "monetary policy committee", "macro-prudential framework", "aggregate financing to the real economy" (社会融资规模) are not in general-purpose dictionaries; one mis-split changes the head word.
- **Fix:** load your domain glossary into the tool's **user dictionary** so policy terms survive as whole tokens. Stopwords differ too: drop structural particles (的, 了, 是); keep logical connectives (因此 "therefore", 然而 "however") for information-extraction tasks.

Preprocessing Is a Modeling Decision

Every cleaning switch changes the downstream estimate

Tokenization Is a Preprocessing Choice

- Tokenization sits inside a larger **preprocessing pipeline**:

$$x^{\text{raw}} \xrightarrow{\text{clean}} x^{\text{clean}} \xrightarrow{\text{tokenize}} \{t_{ij}\} \xrightarrow{\text{represent}} z_i,$$

  from raw scraped text → normalized text → integer token IDs → downstream numeric representation (TF-IDF, embedding, ...).

The Seven Preprocessing Choices

- **Lowercasing** — map `Fed` and `fed` to the same token. Compresses vocabulary, but erases proper-noun signal.
- **Stemming / lemmatization** — cut suffixes so *banking/banks/banked* collapse to *bank*. Can erase tense, plurality, policy-language nuance.
- **Stop-word removal** — drop high-frequency function words. Reduces noise; but negation (*not*) and modality (*may, will*) can matter for central-bank text.
- **Punctuation handling** — keep, strip, or tokenize separately. Percent signs, decimals, sentence boundaries all matter.
- n**-gram inclusion** — treat *"yield curve"* as one token rather than two. Captures multi-word economic concepts at the cost of a larger vocabulary.
- **Frequency cutoffs** — drop terms in $< k$ or $> K$ documents. Removes typos and boilerplate; can also remove rare-but-informative events.
- **Language filtering** — restrict to English or apply per-language pipelines. Crucial for multilingual corpora like ECB speeches.

One Token Boundary Can Flip the Sign

Where you draw token boundaries can reverse an economic conclusion.

FOMC-style sentence: *"...growth is **not sufficient** to warrant further tightening."*

- **Strategy A (unigrams):** `not` and `sufficient` are separate features. If `sufficient` loads positively in training, a bag-of-words model can extract a *positive* signal from a negation.
- **Strategy B (bigrams):** `not_sufficient` is one feature — the negation survives as a unit.

- Does the choice still matter under BERT-class models, whose attention can connect `not` to `sufficient` across positions? (Preview: mostly no — one reason Acts 3–4 exist.)
- Either way: the tokenizer version and configuration belong in the replication package — they are part of the measurement instrument.

Bag-of-Words and TF-IDF

The transparent baseline: count, weight, audit

Bag-of-Words: The Document–Term Matrix

The simplest representation: count words, forget order.

- Toy corpus — doc 1: *"central bank cuts rates to support growth"*; doc 2: *"central bank raises rates to curb inflation"*.

	central	bank	cuts	raises	rates	support	curb	growth	inflation
doc 1	1	1	1	0	1	1	0	1	0
doc 2	1	1	0	1	1	0	1	0	1

- In general: D documents $\times$ vocabulary of size V (typically $V \gg 10^4$); entry $X_{d,v}$ is binary, a raw count, or a TF-IDF weight (two slides ahead).
- **The assumption you buy:** "central bank cuts rates to support growth" = "growth supports central bank rate cuts". Harmless when counting topic words (EPU); fatal for direction, causation, negation.

Bag-of-Words: Two Built-in Consequences

Counting words and forgetting order leaves two statistical fingerprints on the matrix.

- **Sparsity:** a document uses only 1–2% of the vocabulary $\Rightarrow$ the matrix is $>$99% zeros; with $D \ll V$, OLS on raw counts is not even identified — dimension reduction (PCA, topic models) or regularization (Lasso) is mandatory.
- **Orthogonality:** every word is its own perpendicular axis. *Unemployment* $\perp$ *joblessness*: two articles about the same phenomenon in different vocabulary register as *unrelated*.

Orthogonality is the recall problem behind every dictionary method — and the motivation for the dense embeddings of T2b.

Term Frequency and Inverse Document Frequency

Weight words by how much they distinguish a document — not by how often they appear.

TF — share of document d that is word w:

$$\mathrm{TF}(w,d) = \frac{\mathrm{count}(w,d)}{\text{total words in } d}$$

"the Fed raised rates because inflation persisted" (7 words): every word gets $\mathrm{TF} = 1/7 \approx 0.14$ — `the` scores as high as `inflation`.

Limitation: TF alone rewards frequency, not information. We need a counterweight.

IDF — rarity of w across N documents:

$$\mathrm{IDF}(w) = \log\left(\frac{N}{\#\{\text{docs containing } w\}}\right)$$

Corpus of $N = 100$ FOMC speeches:

word	docs	IDF
the	100	0
inflation	60	0.51
yield curve	5	3.00

Rare ⇒ informative ⇒ up-weighted. *the* zeroes out automatically — no stop-word list needed.

TF-IDF — and When Counting Still Wins

Multiply term frequency by inverse document frequency; each document becomes a sparse vector in $\mathbb{R}^{|V|}$.

$$\text{TF-IDF}(w, d) \;=\; \text{TF}(w, d) \cdot \text{IDF}(w)$$

Orthogonality remains a real defect — yet counts win whole classes of research problems:

- **When the target is "how often do well-defined words appear":** EPU (Thread A), the Loughran–McDonald finance dictionary, Hassan et al.'s bigram political risk — all count-based *by design*.
- **The killer feature is auditability:** you can list exactly which article and which phrase triggered the index — line-by-line external validation. No embedding offers that evidence chain.
- **Also the sensible default when:** the corpus is small ($D < 1{,}000$); compute is tight; or the measure must plug into a standard econometric pipeline with transparent features.

Choose the representation for the *role* the text plays (T1's three roles): measurement wants transparency first; behavior and nuance want context — exactly where T2b and Act 3 head next.

Bridge: Counts Cannot See Synonyms

The bag of words has taken us as far as orthogonal axes can go.

- In the document–term matrix, *hawkish* and *tightening* are perpendicular; *unemployment* and *joblessness* share nothing. Every count-based measure inherits this blindness — the recall problem behind all dictionary methods.
- **Next deck:** learn coordinates *from co-occurrence*, so related words end up near each other — Word2Vec, GloVe, and the geometry that lets us do arithmetic with meaning.
- The DGP lens carries over unchanged: an embedding is just a richer $m(\cdot)$; the hazards a_i, c_i do not vanish because the vectors get denser.

*Arc: T1 FRAME → **T2 REPRESENT** → T3 ATTEND → T4 PRETRAIN → T5 MEASURE.*