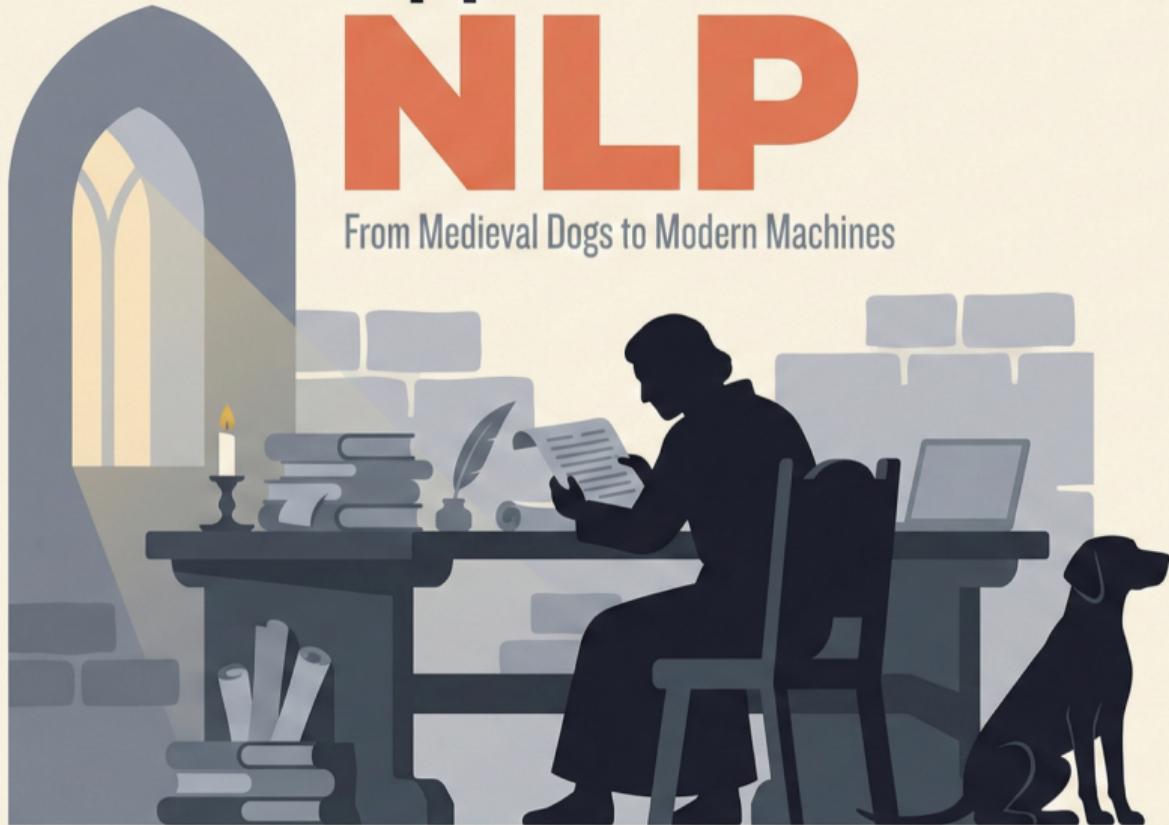


# Applied **NLP**

From Medieval Dogs to Modern Machines



# Two Tracks

Concepts first, systems second

## Concepts & Intuition

We build mental models before code:

- representation (counts → vectors → context)
- matching and retrieval (what is relevant?)
- structure (how we turn text into analyzable data)

Goal: understand why each method works and fails.

## Operations on Text

Treat text as a universal interface to documents, code, logs, and speech transcripts.

**Classify** → assign labels

**Extract** → schema/JSON fields

**Retrieve** → evidence at scale

**Reason** → synthesize + decide

We move from simple pipelines to advanced systems.

# Roadmap

Methods evolve, and so do capabilities



## Capability unlocks

- Better representation of meaning
- Better handling of context and ambiguity

## What scales now

- extraction + labeling + retrieval at corpus scale
- hybrid workflows (classical + neural + LLM)

Each era adds tools; good systems combine them rather than replacing everything.

1

# TF-IDF & Text Features

---

# Word Counts & Information Theory

## The foundation of text retrieval

- Text is **unstructured** — the hardest data type for machines
- First idea: just **count words**
- Problem: common words (“the”, “is”, “and”) dominate
- Solution: weight words by how **informative** they are

### Information-Theory Lens

Self-information is  $I(x) = -\log p(x)$ : rare events carry more bits.

IDF is mathematically equivalent to **self-information (surprisal)**.

In text: rarer terms are often more discriminative.

Shannon, 1948; Sparck Jones, 1972; Aizawa, 2003

# Bag of Words

The simplest text representation

	climate	tax	merger	trial	chatbot	patent
“Climate patent trial update”	1	0	0	1	0	1
“Tax merger debate”	0	1	1	0	0	0
“Chatbot climate memo”	1	0	0	0	1	0

- Each document = a vector of word frequencies
- Ignores word order (“dog bites man” = “man bites dog”)
- But surprisingly effective for classification and retrieval

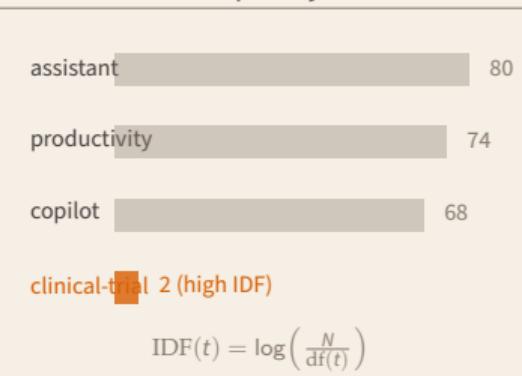
# TF-IDF Intuition: Signal vs. Noise

What is distinctive in a noisy stream?

Imagine 100 AI company announcements this month.

- Most announcements repeat generic terms: **assistant, productivity**
- Those words are everywhere, so they barely distinguish documents
- One announcement includes **clinical-trial evidence**
- That rare term is a strong pointer to a specific sub-topic (health/biotech)

Document frequency in 100 docs



TF-IDF boosts words that are frequent *in this document* but rare *in the corpus*.

# TF-IDF: The Math

$$w(t, d) = \underbrace{\text{tf}(t, d)}_{\text{term frequency}} \times \underbrace{\log\left(\frac{N}{\text{df}(t)}\right)}_{\text{inverse document frequency}}$$

Term	TF in doc	DF (of 1000)	IDF	TF-IDF
“assistant”	10/100	980	$\log(1.02) \approx 0.02$	0.002
“productivity”	7/100	720	$\log(1.39) \approx 0.33$	0.023
“clinical”	3/100	35	$\log(28.6) \approx 3.35$	0.101
“oncology”	2/100	8	$\log(125) \approx 4.83$	0.097

Common terms get low weight; domain-specific rare terms dominate TF-IDF signal.

# TF-IDF: A Worked Example

## Numerical intuition with one announcement

Suppose one document has 80 tokens and discusses a clinical AI launch.

Term	TF	DF/1000	IDF	TF-IDF	Interpretation
assistant	8/80	950	0.05	0.005	very common term, weak signal
productivity	5/80	700	0.36	0.023	moderately informative
clinical	3/80	35	3.35	0.126	rare and highly distinctive
oncology	2/80	8	4.83	0.121	very rare, high discrimination

Key idea: a term can appear fewer times but still dominate if it is globally rare.

# BM25: TF-IDF's Descendant (Still Alive in 2026)

## The algorithm that powers search engines

- BM25 adds **term frequency saturation**:  $\frac{tf}{tf+k_1}$
- Prevents long documents from dominating
- Default in **Elasticsearch**, Apache Solr, Apache Lucene

## 2024: BMX – The Next Step

BMX combines entropy-weighted similarity with TF-IDF.

Outperforms BM25 on the BEIR benchmark.

See Li et al., 2024 (arXiv:2408.06643).

**Lesson:** foundational methods don't die — they **evolve**.  
Every modern search engine still uses TF-IDF descendants.

2

# Topic Modeling

---

# Uncovering Hidden Relationships

## From words to latent structure

### LSA / LSI

SVD on term-document matrix.  
Captures global co-occurrence.  
Deerwester et al., 1990

### LDA

Probabilistic generative model.  
Documents as topic mixtures.  
Blei et al., 2003

### NMF

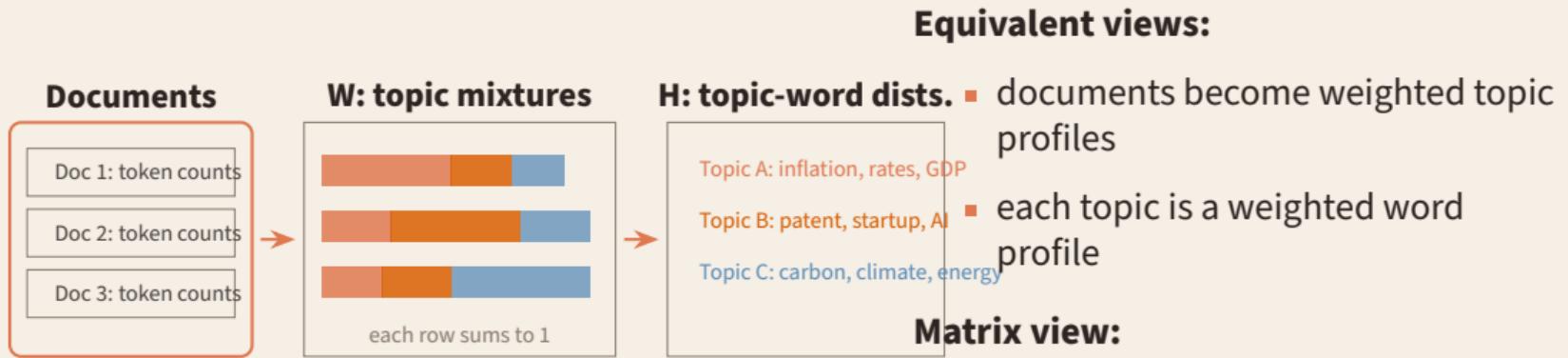
Non-negative factorization.  
Usually easier to interpret.  
Lee & Seung, 1999

- **Method evolution:** algebraic compression → probabilistic semantics → interpretable factorization.
- **In social science:** LDA became a workhorse, but evaluation remains partly subjective.
- Practical rule: judge quality by coherence, usefulness for the question, and human interpretability.

Deerwester et al., 1990; Blei et al., 2003; Lee & Seung, 1999

# Matrix Factorization

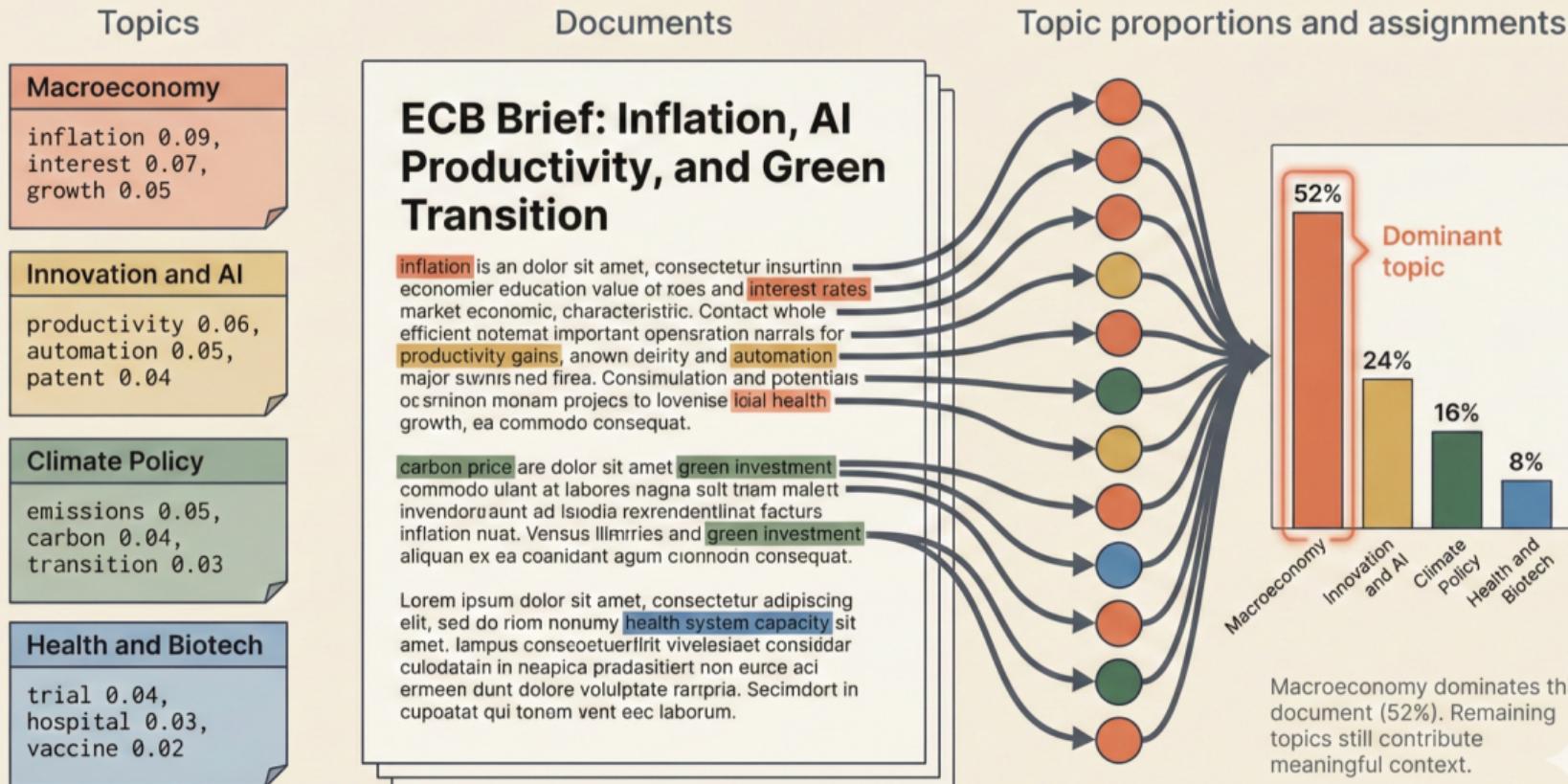
An intuitive map: documents → topics → words



$$A_{\text{docs} \times \text{words}} \approx W_{\text{docs} \times \text{topics}} \cdot H_{\text{topics} \times \text{words}}$$

- **W:** “how much of each topic in each document”
- **H:** “which words define each topic”

# Topic Modeling: Policy Text to Topic Proportions



# Discovering Topics in Text

Automatic theme extraction from a large corpus

Feed 10,000 documents into a topic model → four coherent themes:

## Technology

AI, startup, funding, platform, compute, deploy

## Climate

emissions, carbon, renewable, ESG, transition, green

## Economics

inflation, GDP, labor, trade, policy, fiscal

## Health

vaccine, trial, patients, drug, biotech, FDA

An article on “AI for drug discovery”: 50% Technology, 30% Health, 20% other  
Documents are mixtures, not single labels.

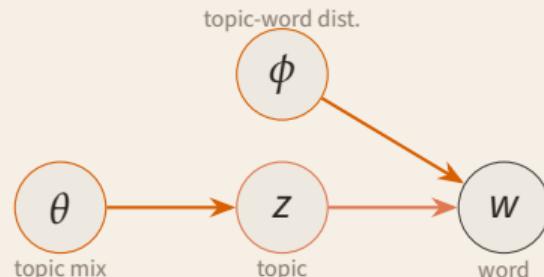
# LDA: The Generative Model

Formal process + plain-language intuition

## Generative story

1. Draw document topic mix:  $\theta_d \sim \text{Dir}(\alpha)$
2. For each token position  $n$ , sample a topic:  
 $z_{d,n} \sim \text{Mult}(\theta_d)$
3. Sample a word from that topic:  
 $w_{d,n} \sim \text{Mult}(\phi_{z_{d,n}})$

Intuition: pick a theme, then pick a word typical for that theme.

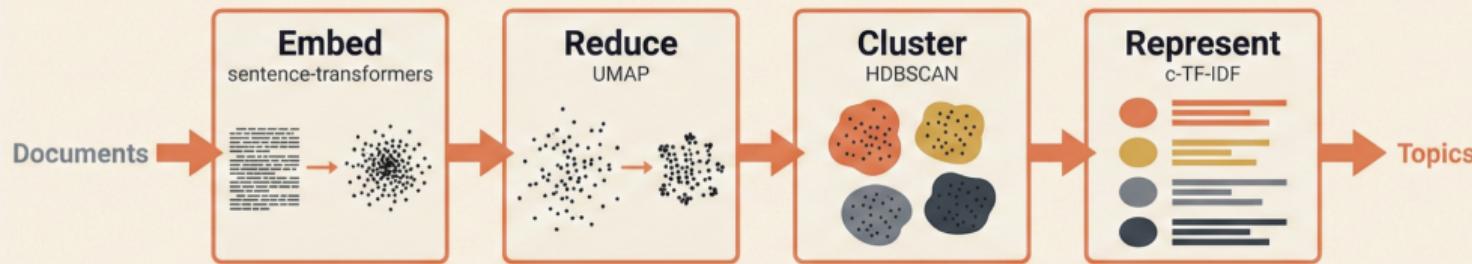


Example token flow:  
[Topic=Health] → vaccine  
[Topic=Economics] → inflation

Blei, Ng & Jordan, 2003

# BERTopic: Topic Modeling for 2026

Embeddings + clustering + interpretability



## Advantages over LDA:

- Auto-selects topic count
- 50+ languages
- LLM-assisted topic labels
- Local models run on laptops (privacy + cost)

## Benchmark:

- Coherence (Cv): **0.76** vs LDA 0.38 ( $\sim 2 \times$ )
- Strong open-source ecosystem + notebook tooling
- New releases: multi-GPU and Model2Vec support

# BERTopic in Practice

## Why social scientists love it

- **Interpretable**: c-TF-IDF gives real words per topic (not just topic IDs)
- **Scalable**: handles 100K+ documents on a laptop
- **LLM integration**: feed topic words to GPT/Claude for human-readable names
- **Visualization**: built-in topic maps, hierarchies, temporal trends
- **Local-first option**: strong small/local models can label topics offline

## Bridge to Social Science

BERTopic is the most adopted topic model in social science since LDA.

Used in: innovation studies, policy analysis, media research, scientometrics.

We'll build a full BERTopic pipeline in **NB04**, including a topic map and LLM topic naming.

3

# Word Embeddings & Vector Space

---

# Words as GPS Coordinates

From sparse counts to dense meaning

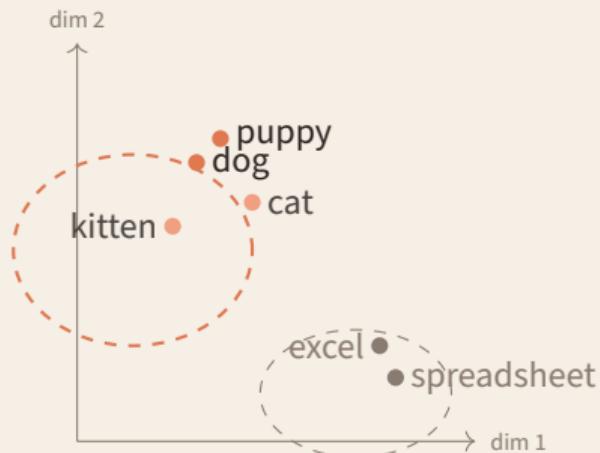
TF-IDF: each word = a dimension (10,000+ dims)

Word2Vec: each word = a **dense vector**  
(100–300 dims)

Think of it as **GPS coordinates in meaning-space**:

- “dog” and “puppy” are **nearby**
- “dog” and “spreadsheet” are **far apart**
- Similar meanings → similar coordinates

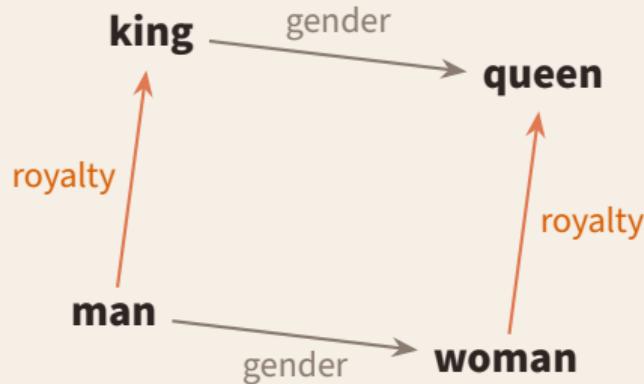
Distributional idea: “you know a word by the company it keeps.” Meaning emerges from context at scale.



# Vector Arithmetic

The most famous equation in NLP

$$\vec{\text{king}} - \vec{\text{man}} + \vec{\text{woman}} \approx \vec{\text{queen}}$$



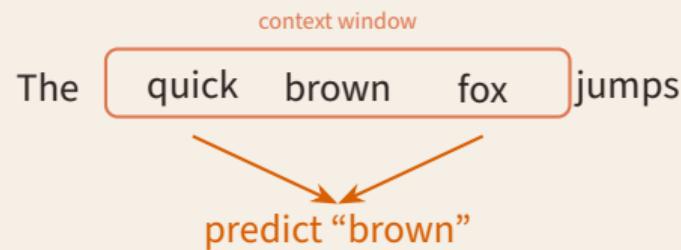
Works for: Paris – France + Germany  $\approx$  Berlin  
Captures relational structure, not just similarity.

# How Word2Vec Learns

## Predicting context from massive text corpora

The model reads thousands of books, predicting words from context:

“The **dog** fetched the [\_\_\_\_\_] from the garden.”



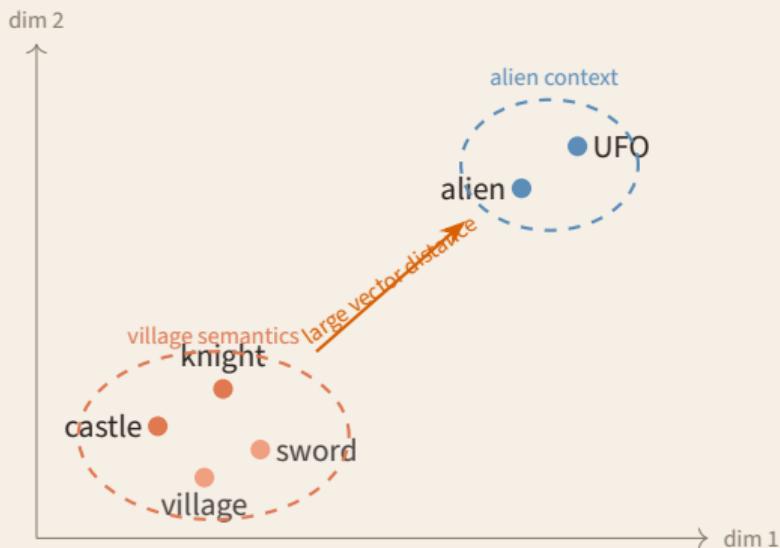
- **Skip-gram**: predict context from center word
- **CBOW**: predict center word from context
- Words that appear in similar contexts get **similar vectors**

# The UFO in the Village

Distance = dissimilarity in vector space

- Frequent co-occurrence pulls vectors together.
- Rare concepts with different context drift far away.
- Distance in embedding space approximates semantic distance.

Distributional hypothesis: “You shall know a word by the company it keeps.” If “UFO” rarely appears with village-life terms, it lands in a different region.



# Visualizing Embedding Space

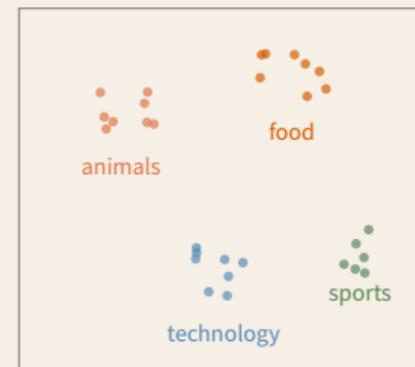
From 300 dimensions down to 2

Dimensionality reduction:

- **t-SNE**: preserves local structure
- **UMAP**: preserves global + local
- Both: 300D → 2D for visualization

What you see:

- Semantic clusters (animals, food, tech)
- Analogies as parallel lines
- Outliers = unusual words



t-SNE / UMAP projection

# Bias in Embeddings

## A warning for social scientists

- Word2Vec trained on Google News:
  - “man” → “computer programmer”
  - “woman” → “homemaker”
- Embeddings **absorb and amplify** societal biases from training data
- WEAT test: measures stereotypes in embedding space

### For Social Scientists

This is both a **bug** and a **feature**:

- Bug: biased models produce biased outputs
- Feature: embeddings **measure cultural associations at scale**

# From Words to Sentences

## Sentence-BERT and the MTEB era

- Word2Vec: one vector per *word*
- Sentence-BERT: one vector per sentence/paragraph
- **MTEB** = Massive Text Embedding Benchmark
- Multilingual embeddings map similar meaning across languages into nearby regions

Practical speedup: pairwise semantic search drops from hours to seconds with bi-encoder embeddings.

### Provider landscape:

- API: OpenAI, Gemini, xAI/Grok (where available)
- Local/open: SBERT, BGE, E5, LLM-derived embeddings
- Ollama + HF pipelines for local-first workflows

### Ops tips:

- Batch embedding APIs for cost/performance
- Cache vectors and version your embedding model

4

# Transformers

---

# Attention Is All You Need

## The paper that changed everything

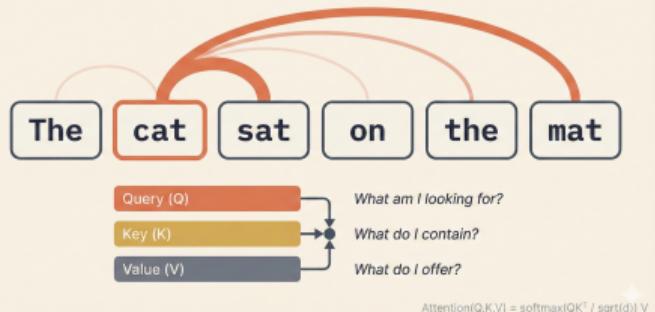
Why attention was needed:

- RNN-style models are sequential and brittle on long-range dependencies
- Attention lets each token directly weigh other tokens
- Computation becomes parallel and more context-rich

**Self-attention:** each token scores all others through Q/K/V.

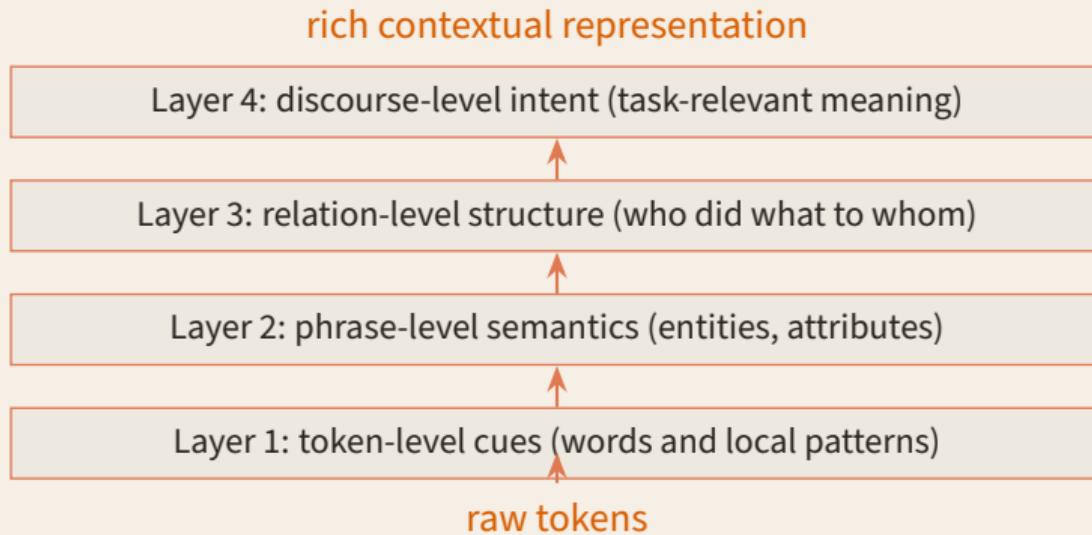
**Query:** what to look for; **Key:** what each token contains; **Value:** what it contributes.

### Self-Attention



# The Playlist Curation Analogy

Each transformer layer adds context



Like curating a playlist: each pass adds signal until the final ranking reflects deeper context.

# Three Transformer Architectures

Architecture	Direction	Models	Tasks
Encoder-only	↔ bidirectional	BERT, RoBERTa	Classification, NER, similarity
Decoder-only	→ left-to-right	GPT, Claude, Llama	Generation, chat, reasoning
Enc-Decoder	↔ + →	T5, BART	Translation, summarization

## BERT: Understanding

Uses bidirectional context via masked-token training.

“I went to the **bank**”

→ river bank? financial bank?

BERT uses *full context* to decide.

## GPT: Generation

Predicts the *next word*.

“Smartphone autocomplete on steroids.”

Counter: “Saying LLMs just predict the next word is like saying a cathedral is just a pile of stones.”

Devlin et al., 2019; Radford et al., 2019; Raffel et al., 2020

# The Evolution of Text Similarity

Same sentences, different representations

	TF-IDF	Word2Vec	SBERT
“Startup raises funding” vs “Company secures capital”	0.12	0.71	0.88
“AI lab cuts costs” vs “Model training becomes cheaper”	0.08	0.66	0.84
“python script failed” vs “python snake escaped”	0.21	0.49	0.09

## TF-IDF

Word overlap only.

“puppy” ≠ “dog”

## Word2Vec

Semantic similarity.

“puppy” ≈ “dog”

## SBERT

Contextual meaning.

Disambiguates “bank”

5

# 2025/2026 State of the Art

---

# The LLM Landscape

February 2026

OpenAI

GPT-5.2, o3  
Codex 5.3

Anthropic

Claude Opus 4.7

Google

Gemini 3

Mistral

Large 3

Meta

Llama 4

DeepSeek

V3.2, R1

Qwen

Qwen 3

xAI

Grok

**Key shift:** DeepSeek/Qwen accelerated Chinese AI adoption from 1.2% → 30% global usage share in one year.

Stanford HAI 2025: Chinese developers 17.1% of HuggingFace (vs US 15.8%). 63% of fine-tuned models use Chinese bases.

# The Model Zoo

Key specifications, early 2026

Model	Context	Strength	Input \$/M	Note
GPT-5.2	400K	Reasoning	\$1.75	100% AIME
Claude 4.5 Sonnet	200K	Coding	\$3.00	77% SWE-bench
Gemini 3 Pro	2M	Multimodal	varies	1501 LMArena Elo
Llama 4 Scout	10M	Open-source	free	17B active/109B
DeepSeek V3.2	128K	Cost	\$0.27	37B active/671B
Qwen 3	128K	Multilingual	free	119 languages

Llama 4 Scout: 10M tokens  $\approx$  15,000 pages in one prompt.

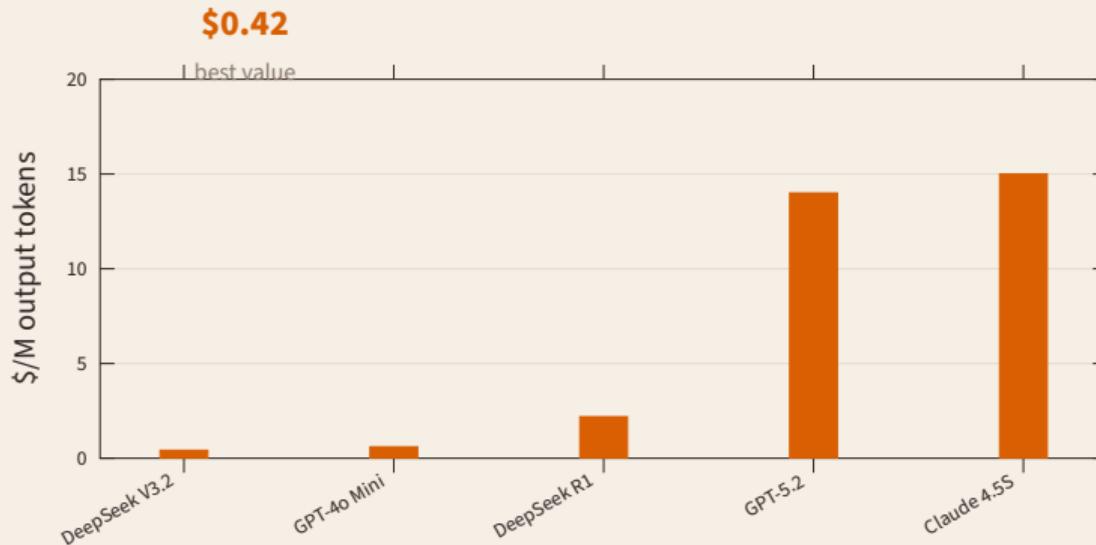
Google: near-perfect needle-in-a-haystack across text, 10.5h video, 107h audio.

needle-in-a-haystack intuition

• needle •

# The Cost of Intelligence is Collapsing

100× cheaper in 2 years



Output tokens cost 3–8× more than input across all providers.

# Reasoning Models: Think Longer, Not Bigger

## The paradigm shift of 2024–2025

**The idea:** spend more compute at *inference time* instead of making models bigger.

- **o1** (Sep 2024): internal chain-of-thought
- **o3** (Apr 2025): 87.5% ARC-AGI
- **QwQ** (Nov 2024): early open reasoning release
- **DeepSeek-R1**: mainstream breakout (not first)

**Key finding** (Snell et al., 2024):

A smaller model with more inference compute outperforms a **14× larger model** that answers instantly.

Visible reasoning:

- OpenAI: hidden CoT
- DeepSeek:  
`<think>...</think>`
- **Transparent** vs hidden

Wei et al., 2022; Snell et al., 2024

# DeepSeek: The Earthquake

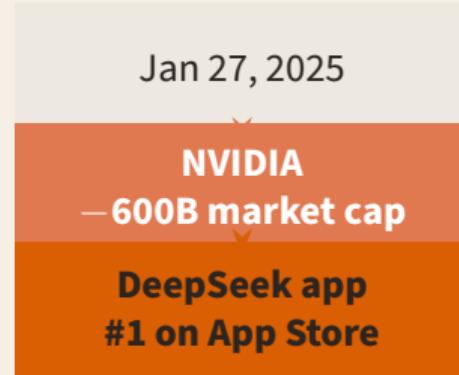
January 2025 — the day NVIDIA lost \$600B

DeepSeek-V3 (Dec 2024):

- MoE: **671B total, 37B active** ( $\sim 5.5\%$ )
- Training cost: **\$5.6M** on 2,048 H800 GPUs
- vs GPT-4 estimated \$50–100M
- Engineers coded in **PTX** (GPU assembly)
- FP8 mixed-precision at extreme scale

DeepSeek-R1 (Jan 2025):

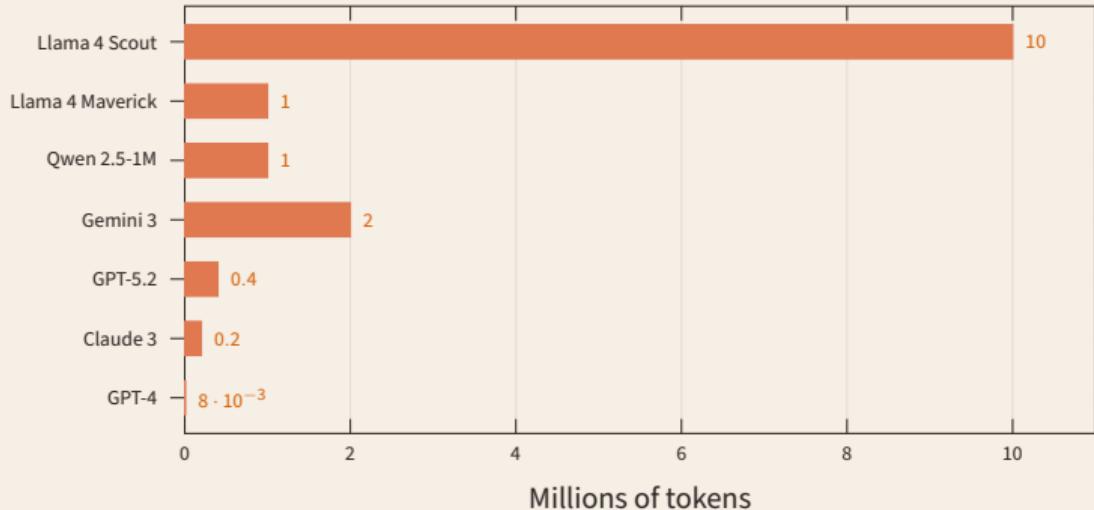
- **Pure RL** (no supervised fine-tuning)
- Matches o1 at **90–96% lower cost**
- **MIT license** — fully open
- 32B distilled model beats o1-mini



“Scarcity fosters innovation”  
— Brookings Institution

# Context Windows in 2026

## From 4K tokens to 10M tokens in 3 years



# Structured Output: LLMs as Data Extraction Engines

Not just chat — structured data

LLMs can return **guaranteed JSON**:

- Constrained decoding
- Pydantic schema validation
- Retry on parse failure

Tools:

- OpenAI Structured Outputs
- `instructor` library (3M+/mo)
- `outlines` (token-level FSM)

```
class ArticleInfo(BaseModel):  
    title: str  
    key_people: list[str]  
    sentiment: Literal[  
        "positive", "negative",  
        "neutral"  
    ]  
    topics: list[str]  
    confidence: float  
  
    # LLM returns valid JSON  
    # matching this schema
```

We'll build this in **NB03** — extracting structured data from news articles.

# The Benchmark Landscape

MMLU is saturated — what's next?

Saturated (90%+ for top models):

- MMLU
- HellaSwag
- ARC-Challenge

Still differentiating:

- **GPQA Diamond**: PhD-level science

Gemini 3: 91.9%

- **AIME**: Math olympiad

GPT-5.2: 100%

- **SWE-bench**: Real GitHub issues

Claude 4.5: 77.2%

The new frontier:

**Humanity's Last Exam (HLE)**

Published in *Nature*, 2025

- 1,000 experts, 500+ institutions
- 2,500 questions, 100+ subjects
- Jan 2025: top models <10%
- Feb 2026: **Gemini 3 Pro: 37.2%**
- GPT-5.2: 35.4%
- Human experts: ~90%

# Multimodal AI

Text, images, video, audio — in one model

Vision foundations:

- **CLIP** (2021): text + images in same vector space
- Vision-Language Models: GPT-4o, Gemini, Claude see images natively
- Open: Qwen2.5-VL, LLaMA 3.2 Vision

Image generation:

- FLUX, GPT Image 1.5, Midjourney V7
- Accurate text in images (Ideogram 3.0)

Video generation (2025–26):

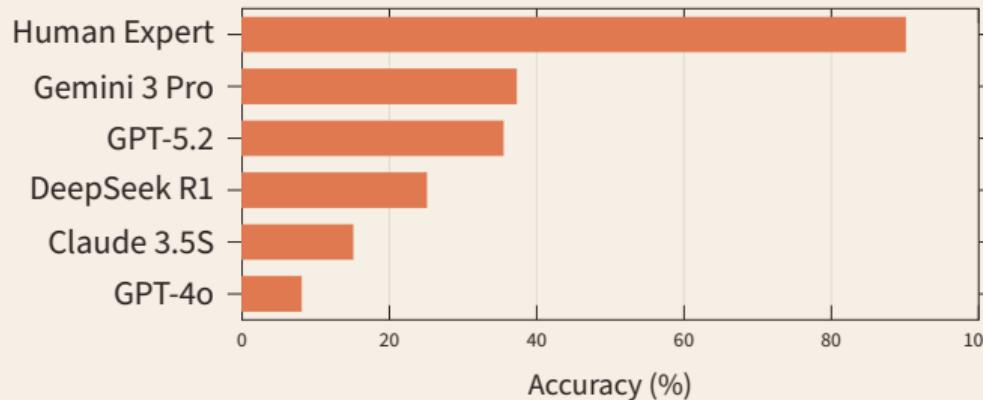
- Sora 2: longer clips + synchronized audio
- Kling 3.0: native 4K 60fps demos
- Veo 3.1 and WAN 2.6 (open)

Audio/Speech:

- Qwen3-TTS: strong open speech synthesis
- Kokoro: lightweight local TTS
- Whisper large-v3: robust ASR
- ElevenLabs v3: multilingual voice generation

# Humanity's Last Exam

1,000 experts. 100+ subjects. Best AI still below 40%.



## Coding frontier (2026)

Coding evals (SWE-bench style and GPT-Val variants) are now key differentiators; practical software engineering remains a live benchmark race.

6

# Applied Social Science

---

# NLP for Economists & Social Scientists

## Text as Data

- **Gentzkow, Kelly & Taddy** (2019): “Text as Data” — canonized the field
- **Ash & Hansen** (2023): first major econ survey on embeddings + transformers
- Social science is adopting NLP with a **4-year diffusion lag**

### What NLP enables for social science:

- **Scale**: analyze 100,000+ documents (policy, patents, speeches)
- **Measurement**: cultural dimensions from text (Kozlowski et al., 2019)
- **Replication**: LLMs outperform crowd-workers for annotation (Gilardi et al., 2023)
- **Simulation**: “*Homo Silicus*” — LLMs as simulated survey respondents (Horton, 2023)

# Embeddings for Innovation Research

## Measuring technological change with text

**Patent similarity** (Arts et al., 2018, 2021):

TF-IDF + cosine on patent text to measure technological relatedness

**Breakthrough detection** (Kelly et al., 2021):

A patent is “important” if *textually distant* from prior work but *similar to subsequent*.

Covers 1840–present!

**PatentSBERTa** (Bekamiri, Hain & Jurowetzki, 2024):

Fine-tuned SBERT on patent pairs for semantic patent matching

NLP + economic geography: map knowledge spaces and identify smart-specialization opportunities.



# AI & Productivity: The Evidence

## Three landmark experiments

BCG + Harvard  
Dell'Acqua et al., 2023

758 consultants

+25% faster

+40% quality

within AI's frontier

-19% quality

outside frontier

Lowest performers:  
**+43% improvement**

MIT / Science  
Noy & Zhang, 2023

453 professionals

-40% time

+18% quality

Greatest benefits for  
**lower-ability workers**

Published in *Science*

Stanford / QJE  
Brynjolfsson et al., 2023

5,172 CS agents

+14% overall

+34% for novices

AI **disseminates tacit knowledge**  
of top performers

Published in *QJE*

Pattern: AI is an *equalizer* — biggest gains for least experienced workers.

# The Adoption Gap

Social science is catching up — fast

## Why the lag?

- Interpretability requirements
- Causal identification culture
- Smaller datasets / qualitative traditions
- Institutional inertia

## Why it's closing:

- BERTopic: interpretable by design
- Structured output: LLM → DataFrame
- Cost collapse: everyone can afford it
- Embedding-based measurement at scale

## The Jagged Frontier (Dell'Acqua et al.)

AI excels at some tasks, fails at others — and the boundary is jagged.  
The professional skill is knowing *when* to use AI and when not to.

# Who Is Affected?

## AI task exposure across the economy

Eloundou et al. (2024, *Science*):

- ~80% of US workforce:  $\geq 10\%$  of tasks affected
- ~19%:  $\geq 50\%$  of tasks affected
- Higher-income jobs face greater exposure

### Task Exposure Snapshot

80% with  $\geq 10\%$  tasks

19% with  $\geq 50\%$  tasks

Exposure rises with income

Acemoglu (2024 Nobel laureate):

“The Simple Macroeconomics of AI”

Estimate:  $\leq 0.66\%$  TFP increase over 10 years

Modest macro effect, but large for specific tasks and workers.

# Workshop Preview & Closing

---

# What We'll Build This Week

## Outcomes, systems, and evaluation

### Day 1: Foundations & Scale

- TF-IDF baselines
- Embedding pipelines
- BERTopic mapping
- Vector similarity
- + first retrieval checks

### Day 2: The Retrieval Engine

- Vector DBs
- FAISS + reranking
- RAG design
- Eval + error analysis
- + retrieval QA loop

### Day 3: Production Agents

- Structured extraction
- Distillation + tuning
- Agentic workflows
- Guardrails + handoff
- + demos + wrap-up



Build, compare, and validate: each day turns methods into deployable workflows.

# The Professional's Playbook

Start simple, escalate only when needed

try this first

1. **Prompting** — can a good prompt solve it?
2. **RAG** — does it need external knowledge?
3. **Few-shot / SetFit** — can 8–32 examples help?
4. **Fine-tuning** — do you have enough data?
5. **Custom training** — is this a novel task?

complexity & cost

Most real-world NLP problems are solved at levels 1–2.

# Five Themes to Remember

## 1. Foundational methods haven't been replaced

TF-IDF lives in BM25. Embeddings are the backbone of search.

## 2. The reasoning revolution

Inference-time compute and transparent reasoning changed model behavior.

## 3. The cost of intelligence is collapsing

Frontier-level capability is becoming dramatically cheaper year by year.

## 4. Local, privacy-first NLP is now practical

Strong laptop-grade models make private analysis feasible for many teams.

## 5. The adoption gap is closing

Social science is still behind CS, but adoption is accelerating with usable tooling.

# Resources & Further Reading

## Textbooks:

- Grimmer, Roberts & Stewart (2022)  
*Text as Data* (Princeton UP)
- Jurafsky & Martin  
*SLP* 3rd ed. (free online)
- Raschka (2024)  
*Build an LLM From Scratch*

## Visual guides:

- Jay Alammar's Illustrated Series
- 3Blue1Brown Transformer videos
- HuggingFace LLM Course

## Key papers for social science:

- Gentzkow et al. (2019) — JEL
- Ash & Hansen (2023) — Ann. Rev. Econ.
- Kozlowski et al. (2019) — ASR
- Gilardi et al. (2023) — annotation
- Dell'Acqua et al. (2023) — productivity

## This course:

[github.com/RJuro/unistra-nlp2026](https://github.com/RJuro/unistra-nlp2026)  
[rjuro.github.io/unistra-nlp2026](https://rjuro.github.io/unistra-nlp2026)

# References (Round 1 Core) I

-  Aizawa, A. (2003).  
An information-theoretic perspective of tf-idf measures.  
*Information Processing & Management*, 39(1):45–65.
-  Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003).  
Latent dirichlet allocation.  
*Journal of Machine Learning Research*, 3:993–1022.
-  Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990).  
Indexing by latent semantic analysis.  
*Journal of the American Society for Information Science*, 41(6):391–407.
-  Lee, D. D. and Seung, H. S. (1999).  
Learning the parts of objects by non-negative matrix factorization.  
*Nature*, 401(6755):788–791.
-  Li, Y., Ma, X., Ai, Q., and Croft, W. B. (2024).  
Bmx: Entropy-weighted similarity and semantic-aware term importance for lexical retrieval.  
*arXiv preprint arXiv:2408.06643*.

## References (Round 1 Core) II

-  Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013).  
Efficient estimation of word representations in vector space.  
*arXiv preprint arXiv:1301.3781*.
-  Robertson, S. and Zaragoza, H. (2009).  
The probabilistic relevance framework: Bm25 and beyond.  
*Foundations and Trends in Information Retrieval*, 3(4):333–389.
-  Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. (1995).  
Okapi at trec-3.  
In *Proceedings of the Third Text REtrieval Conference (TREC-3)*, pages 109–126.
-  Shannon, C. E. (1948).  
A mathematical theory of communication.  
*Bell System Technical Journal*, 27(3):379–423.
-  Sparck Jones, K. (1972).  
A statistical interpretation of term specificity and its application in retrieval.  
*Journal of Documentation*, 28(1):11–21.

## References (Round 1 Core) III



Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017).  
Attention is all you need.  
In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, pages 5998–6008.

# Let's begin.

---

20 hours to build your NLP toolkit.

Roman Juowetzki — [rjuro@business.aau.dk](mailto:rjuro@business.aau.dk)