Applied NLP

A Practical Journey Through NLP

Roman Jurowetzki

Aalborg University / University of Strasbourg

February 2026

Two Tracks

How this lecture works

Concepts & Intuition

Visual analogies and worked examples that make the algorithms stick.

From simple counts to deep learning.

Word Counting → TF-IDF
Hidden Topics → Topic Models
Meaning as Geometry → Embeddings
Context is Everything → Transformers

The State of the Art

Architecture diagrams, benchmarks, code snippets, cost tables.

Where we actually are in Feb 2026.

Transformers → Attention
Reasoning Models → o3, R1
Cost Collapse → $0.27/M tokens
Social Science → Text as Data

Roadmap

Teaching Computers Language

TF-IDF	Topics	Word2Vec	Transformers	LLMs	State of Art
1990s	2003	2013	2017	2022+	2026
Word counts	LDA, NMF	Dense vectors	Attention	GPT, Claude	Where we are

Each era added a tool — none replaced what came before.

TF-IDF still powers Elasticsearch. Embeddings are still the backbone of search.

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

Insight for Economists

IDF is mathematically equivalent to self-information (surprisal).
Rare events carry more information — just like in information theory.

Sparck Jones, 1972; Aizawa, 2003

Bag of Words

The simplest text representation

	scary	long	good	funny	boring	great
"Scary and long movie"	1	1	0	0	0	0
"Good and funny film"	0	0	1	1	0	0
"Not a great movie"	0	0	0	0	0	1

- 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: The Dog Park

Finding a unique dog in a crowded park

Imagine a dog park with 100 dogs.

- 80 are **Golden Retrievers**
- “Golden Retriever” is not a useful descriptor
- But **one** dog wears a funny bandana
- That bandana is extremely informative!

TF-IDF = how often a word appears in *this* document
× how *rare* it is across *all* documents

TF-IDF: The Math

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

Term	TF in doc	DF (of 1000)	IDF	TF-IDF
"the"	10/100	1000	$\log(1) = 0$	0
"dog"	8/100	900	$\log(1.1) \approx 0.05$	0.004
"park"	5/100	50	$\log(20) \approx 1.3$	0.065
"bandana"	2/100	3	$\log(333) \approx 2.5$	0.050

"The" appears everywhere $\rightarrow$ zero weight. "Park" and "bandana" are distinctive.

TF-IDF: A Worked Example

Finding what makes a document unique

A forum with 1,000 advice posts across 8 categories.

"Help with my breakup" mentions "relationship" 5 times.

- TF("relationship") = $5/80 = 0.063$
- But "relationship" appears in 600 of 1,000 posts
- IDF = $\log(1000/600) \approx 0.22 \ldots$ modest

Meanwhile, "ghosting" and "rebound" are rare globally
but frequent in *this* post $\rightarrow$ high TF-IDF

Forum Post #42

"relationship" $\times$ 5 ... common
"ghosting" $\times$ 3 ... rare!
"rebound" $\times$ 2 ... rare!
"the" $\times$ 12 ... useless
"catfishing" $\times$ 1 ... very rare!

TF-IDF finds the distinguishing words

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

The algorithm that powers search engines

- BM25 adds **term frequency saturation**: $\frac{\mathrm{tf}}{\mathrm{tf}+k_1}$
- 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.
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

What if documents share latent themes?

- TF-IDF treats each word independently
- But documents have hidden topics that connect words
- **Goal:** discover these topics automatically

LSA / LSI	LDA	NMF
SVD on term-doc matrix (Deerwester, 1990)	Generative model (Blei et al., 2003)	Non-negative factors (Lee & Seung, 1999)

Matrix Factorization

The mathematical trick behind topic modeling

A	$\approx$	W	$\times$	H
docs × words		docs × topics		topics × words

- **W** tells us: each document's topic mixture
- **H** tells us: each topic's word distribution
- We choose *k* topics — the model finds the rest

Discovering Topics in Text

Automatic theme extraction from a large corpus

Feed 10,000 news articles to a topic model → 4 hidden themes emerge:

Technology	Economics	Climate	Health
AI, startup, funding, platform, compute, deploy	inflation, GDP, labor, trade, policy, fiscal	emissions, carbon, renewable, ESG, transition, green	vaccine, trial, patients, drug, biotech, FDA

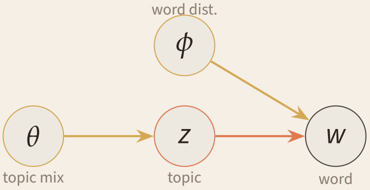
An article on "AI for drug discovery": 50% Technology, 30% Health, 20% other

Documents are *mixtures* of topics — not just one.

LDA: The Generative Model

How documents are “born” according to LDA

1. For each document, draw a topic mixture $\theta \sim \mathrm{Dir}(\alpha)$
2. For each word position:
   2.1 Choose a topic $z \sim \mathrm{Mult}(\theta)$
   2.2 Choose a word $w \sim \mathrm{Mult}(\phi_z)$



Blei, Ng & Jordan, 2003

BERTopic: Topic Modeling for 2026

Embeddings + clustering + interpretability

Embed (sentence-transformers) → Reduce (UMAP) → Cluster (HDBSCAN) → Represent (c-TF-IDF)

Advantages over LDA:

- Auto-determines topic count
- 50+ languages
- LLM-powered topic naming
- Dynamic & hierarchical models

Benchmark:

- Coherence (Cv): 0.76
  vs LDA's 0.38 — nearly 2×
- v0.17+: multi-GPU, Model2Vec
- Active development by Maarten Grootendorst

Grootendorst, 2022; arXiv:2203.05794

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

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 — with LLM topic naming via Groq.

3

Word Embeddings & Vector Space

Words as GPS Coordinates

From sparse counts to dense meaning

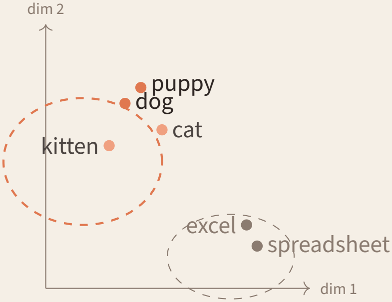
TF-IDF: each word = a dimension (10,000+ dims)

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

Trained on billions of words—the model learns meaning from *context*.

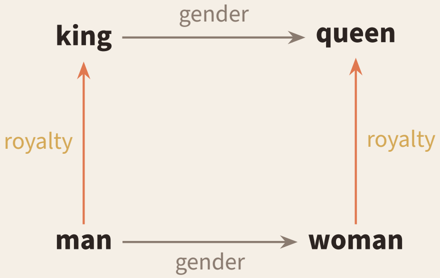


Mikolov et al., 2013

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 ≈ Berlin

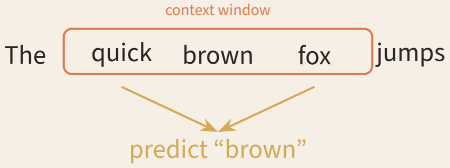
Captures relational structure, not just similarity

How Word2Vec Learns

Predicting context from massive text corpora

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

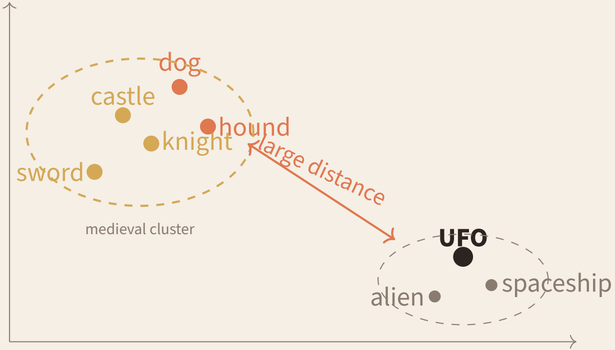
"The dog fetched the [_____] from the garden."



- **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



Concepts that never co-occur end up far apart in vector space.

A UFO landing in a medieval village — clearly out of place!

Visualizing Embedding Space

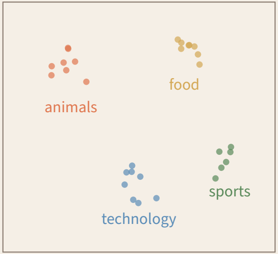
From 300 dimensions down to 2

Dimensionality reduction:

- **t-SNE**: preserves local structure
- **UMAP**: preserves global + local
- Both: 300D $\rightarrow$ 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" $\rightarrow$ "computer programmer"
  - "woman" $\rightarrow$ "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

Bolukbasi et al., 2016; Caliskan et al., 2017; Kozlowski et al., 2019

From Words to Sentences

Sentence-BERT and the MTEB era

- Word2Vec: one vector per *word*
- Sentence-BERT (2019): one vector per *sentence/paragraph*
- Key stat: finding most-similar pair from **65 hours** (BERT cross-encoder) to 5 seconds

Model	Dims	MTEB	Cost
Cohere embed-v4	1024	65.2	$0.10/M tok
OpenAI text-embedding-3-large	3072	64.6	$0.13/M tok
all-MiniLM-L6-v2	384	—	Free
BGE-M3 (multilingual)	1024	—	Free

+ Matryoshka embeddings: truncate dimensions without retraining (Kusupati et al., 2022)

Reimers & Gurevych, 2019

4

Transformers

Attention Is All You Need

The paper that changed everything

Self-attention: every word looks at every other word simultaneously.

- **Query**: what am I looking for?
- **Key**: what do I contain?
- **Value**: what do I offer?

Analogy: Query = your search text,
Key = the page title,
Value = the page content

attention weights

The cat sat on the mat

“cat” attends strongly to “sat” and “mat” — learns relationships.

Vaswani et al., 2017

The Assembly Line Analogy

Each transformer layer adds context

rich contextual representation

Layer 4: → described in **Shakespearean** language

Layer 3: → who **murdered** the previous king

Layer 2: → a **Scottish** male ruler

Layer 1: “king” → a male ruler

raw token: “king”

Like an assembly line: each station adds detail. The final product is

Three Transformer Architectures

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

BERT: Understanding

Reads *both directions* simultaneously.
"I went to the **bank**"
$\rightarrow$ 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	GloVe	SBERT
"The dog ran" vs "The cat ran"	0.67	0.85	0.82
"The dog ran" vs "A puppy sprinted"	0.00	0.72	0.89
"Bank of England" vs "River bank"	0.33	0.55	0.12

TF-IDF

Word overlap only.
"puppy" $\neq$ "dog"

GloVe

Semantic similarity.
"puppy" $\approx$ "dog"

SBERT

Contextual meaning.
Disambiguates "bank"

5

2025/2026 State of the Art

The LLM Landscape

February 2026

OpenAI	Anthropic	Google	Mistral
GPT-5.2, o3	Claude 4.5	Gemini 3	Large 3
Meta	DeepSeek	Qwen	xAI
Llama 4	V3.2, R1	Qwen 3	Grok

Key shift: Chinese AI went from 1.2% → 30% of global usage 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

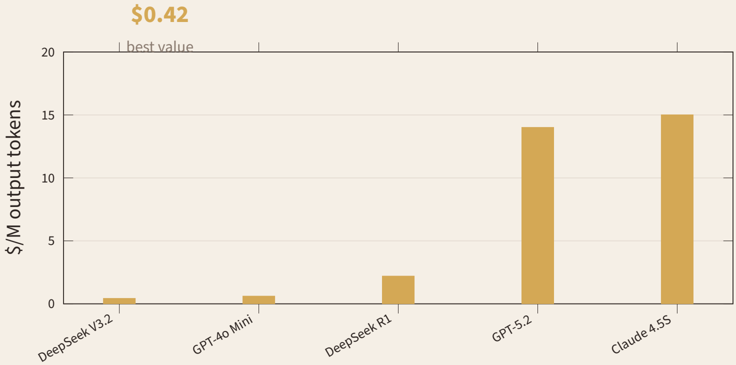
Model	Context	Strength	Input $/M	Note
GPT-5.2	400K	Reasoning	$1.75	100% AIME
Claude 4.5 Sonnet	200K	Coding	$3.00	77% SWE-bench
Gemini 3 Pro	2M	Multimodal	varies	1501 LMArena Elo
Llama 4 Scout	10M	Open-source	free	17B active/109B
DeepSeek V3.2	128K	Cost	$0.27	37B active/671B
Qwen 3	128K	Multilingual	free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

Llama 4 Scout: 10M tokens ≈ **7,500 pages** in one prompt.

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

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
- **DeepSeek-R1**: open-source reasoning

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 (∼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

Jan 27, 2025

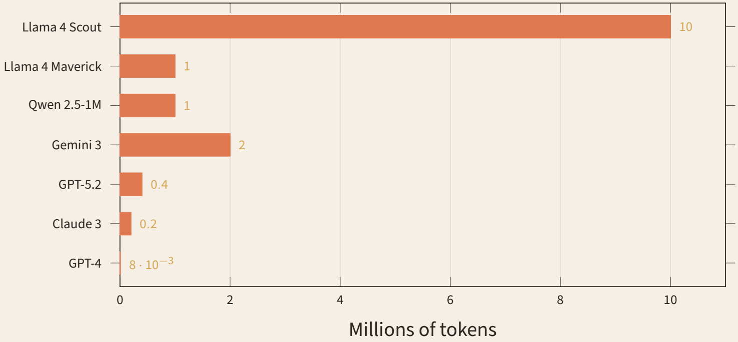
NVIDIA
—600B market cap

DeepSeek app
#1 on App Store

"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)

```python
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: $\sim 90\%$

Hendrycks et al., 2025 (Nature); scale.com/leaderboard

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):

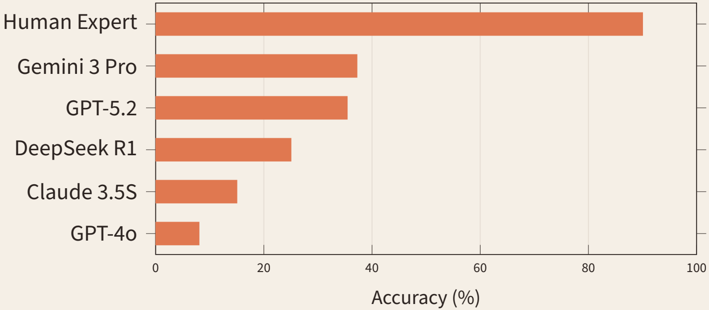
- Sora 2: 25s + synchronized audio
- Kling 3.0: native 4K 60fps
- Veo 3.1: photorealism
- WAN 2.6: open-source

Audio/Speech:

- Whisper large-v3: 1.55B params
- ElevenLabs v3: 32 languages
- NotebookLM: AI podcast from docs
- Real-time speech-to-speech

Humanity's Last Exam

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



AI is superhuman on many tasks — but expert-level knowledge remains hard.

Nature, 2025; arXiv:2501.14249

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)

Gentzkow et al., 2019 (JEL); Ash & Hansen, 2023 (Ann. Rev. Econ.)

Embeddings for Innovation Research

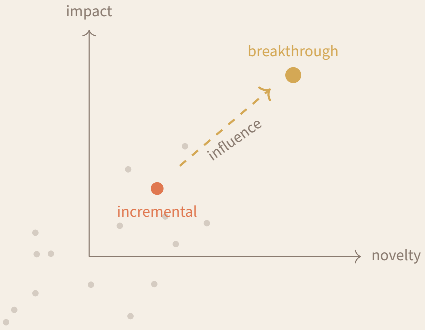
Measuring technological change with text

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, identify



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 $\rightarrow$ 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.

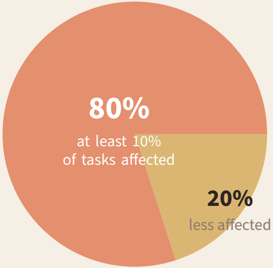
Eloundou et al., 2024 (Science); Acemoglu, 2025 (Economic Policy)

Who Is Affected?

AI task exposure across the economy

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

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



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.

7

Workshop Preview & Closing

What We'll Build This Week

3 days, 11 notebooks, 1 project

Day 1: Baselines

TF-IDF
Embeddings
LLM Zero-shot
BERTopic

+ Sprint 1

→

Day 2: Retrieval

SetFit Few-shot
FAISS Search
Evaluation

+ Sprint planning

→

Day 3: Advanced

Reranking
Distillation
Fine-tuning

+ Sprint 2 + Demos

Every approach compared to the previous one. Error analysis over accuracy chasing.

The Professional's Playbook

Start simple, escalate only when needed

5. **Custom training** — is this a novel task?

4. **Fine-tuning** — do you have enough data?

3. **Few-shot / SetFit** — can 8–32 examples help?

2. **RAG** — does it need external knowledge?

1. **Prompting** — can a good prompt solve it?

try this first

complexity & cost

80% of 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
   o1, o3, R1: "think longer" beats "make bigger."
3. The cost of intelligence is collapsing
   DeepSeek V3.2 at $0.27/M — 100$\times$ cheaper than 2 years ago.
4. The toolkit has matured
   RAG, agents, structured output, fine-tuning, eval — all production-ready.
5. The adoption gap is closing
   Social science is 4 years behind CS — but catching up fast.

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`
`rjuro.github.io/unistra-nlp2026`

Let’s begin.

20 hours to build your NLP toolkit.

Roman Jurowetzki — rjuro@business.aau.dk