

# AI and Unstructured Data for Measurement and Estimation

## Lecture 1: Large Language Models

Stephen Hansen  
University College London



# Background Reading

[Gentzkow et al., 2019a] (pre-LLM methods)

[Ash and Hansen, 2023] (transition period)

[Ash et al., 2025] (LLM methods)

[Ludwig et al., 2025] (LLM methods)

[Jurafsky and Martin, 2025] textbook <https://stanford.io/4qkv65V>

1. Word embeddings, chapter 5

2. LLMs, chapter 7, 8, 10 way beyond most economists would be interested

# Outline

Introduction

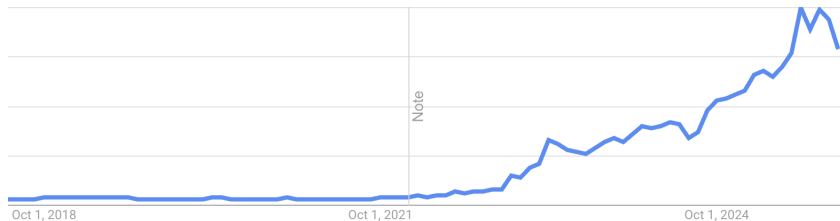
Basic Concepts

Word Embeddings

Attention and Transformer Models

Economic Applications

# Google Trends: “Artificial Intelligence”



# Unstructured Data for Economic Measurement

Typical motivation for using unstructured data in economics is to measure some observation-specific **latent variable**  $\theta_i$ .

Examples include:

1. [Baker et al., 2016]  
 $\theta_i$ : **economic policy uncertainty**; data: newspaper text.
2. [Gentzkow et al., 2019b]  
 $\theta_i$ : **polarization**; data: Congressional Records.
3. [Bandiera et al., 2020]  
 $\theta_i$ : **CEO behavior**; data: time use surveys.
4. [Adukia et al., 2023]  
 $\theta_i$ : **cultural representation**; data: children's book photos.
5. [Gorodnichenko et al., 2023]  
 $\theta_i$ : **Fed chairperson tone**; data: FOMC press conference audio feed.

# How Does AI Change the Landscape?

*Potentially* improves estimation of  $\theta_i$ .

benchmarking exercise is still rare

Reduces the time cost of extracting information from unstructured data.

Brings unstructured data analysis more into the mainstream.

Need to better understand statistical properties of empirical pipelines that incorporate AI.

# Lecture Plan

**Lecture I:** Foundation LLM/AI models.

**Lecture II:** Adapting foundation models for economic measurement.

**Lecture III:** AI/ML-generated estimates  $\hat{\theta}_i$  in regression models.

All material on

[https://github.com/sekhansen/columbia\\_lectures\\_2025](https://github.com/sekhansen/columbia_lectures_2025).

# Outline

Introduction

Basic Concepts

Word Embeddings

Attention and Transformer Models

Economic Applications

# Notation

The corpus is composed of  $D$  documents indexed by  $d$ .

After pre-processing, each document is a finite, length- $N_d$  list of terms  $\mathbf{w}_d = (w_{d,1}, \dots, w_{d,N_d})$  with generic element  $w_{d,n}$ .

Suppose there are  $V$  **unique** terms in the corpus, indexed by  $v$ .

We can then map each term in the corpus into this index, so that  $w_{d,n} \in \{1, \dots, V\}$ .

# Example

Consider three documents:

1. 'stephen is nice'
2. 'john is also nice'
3. 'george is mean'

We can consider the set of unique terms as {stephen, is, nice, john, also, george, mean} so that  $V = 7$ .

Construct the following index:

stephen	is	nice	john	also	george	mean
1	2	3	4	5	6	7

We then have  $\mathbf{w}_1 = (1, 2, 3)$ ;  $\mathbf{w}_2 = (4, 2, 5, 3)$ ;  $\mathbf{w}_3 = (6, 2, 7)$ .

# Document-Term Matrix

In the **bag-of-words model**, we summarize the information in a corpus via term counts.

Let  $x_{d,v}$  be the count of term  $v$  in document  $d$ . The **document-term matrix**  $\mathbf{X}$  collects the counts  $x_{d,v}$  into a  $D \times V$  matrix.

In the previous example, we have

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

each row is one document in  $D$   
each column is one unique word in  $V$

if you start to tabulate subsequence like "growth is week"  
the number of columns will grow very soon

# Limitations of Bag-of-Words

## Synonymy

*economic growth is weak but long-term productivity trends are strong*  
*economic growth is tepid but long-term productivity trends are strong*

# Limitations of Bag-of-Words

## Synonymy

*economic growth is **weak** but long-term productivity trends are strong*  
*economic growth is **tepid** but long-term productivity trends are strong*

## Polysemy

*economic statistics **lie** about current well-being*  
*my cat's favorite activity is to **lie** on our bed*

# Limitations of Bag-of-Words

## Synonymy

*economic growth is weak but long-term productivity trends are strong*

*economic growth is tepid but long-term productivity trends are strong*

solution: PCA on X (latent semantic) to find similar words

this only accounts for the count in the whole document, i.e., global context only, no local

## Polysemy

*economic statistics lie about current well-being*

*my cat's favorite activity is to lie on our bed*

## Sequence

*economic growth is weak but long-term productivity trends are strong*

*economic growth is strong but long-term productivity trends are weak*

the bag of word representation ignore the order or words

although these two sentences have the same bag of words representation, they have different meanings

# Older is Sometimes Better! [Plaza-del-Arco et al., 2024]

Task/Language		best ZSL	supervised	
			Standard ML	Transformer
SA	EN	0.553	0.610	<b>0.680</b>
	DE	0.517	0.610	<b>0.677</b>
	FR	0.528	0.612	<b>0.706</b>
AC-Gender	EN	0.624	0.601	<b>0.638</b>
	DE	0.497	0.540	<b>0.629</b>
	FR	0.579	0.546	<b>0.650</b>
AC-Age	EN	0.572	0.620	<b>0.636</b>
	DE	0.503	0.602	<b>0.611</b>
	FR	0.550	0.540	<b>0.568</b>

standard ML: bag of word, and then logistic model

# Outline

Introduction

Basic Concepts

Word Embeddings

Attention and Transformer Models

Economic Applications

# Language as Vectors

At a very high level, an LLM takes an input text and converts it into a (relatively) low dimensional vector.

This vector might be of interest in its own right, e.g. document similarity.

Or it might be an intermediate representation for a different language task, e.g. predicting an associated label or predicting new words.

LLMs allow the vector representation to change when the word sequence changes to capture meaning.

# Word Embeddings

before vector representation of document,  
first vector representation of word

lion, cat, dog, place these three words in this 2-dimension space  
feline vs canine  
wild vs domestic

A **word embedding** is a low-dimensional vector representation of a word.

Ideally in this low-dimensional vector space words with similar meanings will lie close together.

The construction of word embeddings was an important precursor to the development of large language models.

problem: hard to write out the loss function where meaning is the what we try to capture

The ‘meaning’ of a word is an unobserved and subjective concept.

Difficult to directly formulate an objective function.

Important conceptual idea is to formulate word prediction tasks that are solved using word embeddings.<sup>1</sup>

The approach of using auxiliary word prediction tasks to build high-quality embeddings is called **self-supervised learning**.

---

<sup>1</sup>See also [Bengio et al., 2003].

# Distributional Hypothesis

goal is to build a nn loss function

The **distributional hypothesis** states that words that share similar contexts share similar meanings.

Example from Jurafsky and Martin:

(6.1) Ongchoi is delicious sauteed with garlic.

(6.2) Ongchoi is superb over rice.

(6.3) ...ongchoi leaves with salty sauces...

And suppose that you had seen many of these context words in other contexts:

(6.4) ...spinach sauteed with garlic over rice...

(6.5) ...chard stems and leaves are delicious...

(6.6) ...collard greens and other salty leafy greens

# Formalizing Local Context

The *context* of  $w_{d,n}$  is a length- $2L$  window of words around  $w_{d,n}$ :

$$C(w_{d,n}) = [w_{d,n-L}, w_{d,n-L+1}, \dots, w_{d,n+L-1}, w_{d,n+L}]$$

In line with distributional hypothesis, word embedding models seek to generate similar embeddings for words that share similar contexts.

# Word2Vec

**Word2vec** [Mikolov et al., 2013a, Mikolov et al., 2013b] is a particularly well-known algorithm for the construction of word embeddings.

Important example of a neural-network-based language model that was scalable and effective.

## Skipgram Variant

1. Predict **presence** of each  $w_{d,n-l} \in \mathcal{C}(w_{d,n})$  given  $w_{d,n}$ .
2. Predict **absence** of randomly sampled words from the corpus given  $w_{d,n}$ .

# Words and Context in Skipgram Model

“economic growth is weak but long-term productivity trends are strong”

Suppose  $L = 2$ .

Positive Examples		Negative Examples	
Word	Context	Word	Context
economic	growth	economic	down
economic	is	economic	towards
growth	economic	growth	inflation
growth	is	growth	mild
growth	weak	growth	very
is	economic	is	not
is	growth	is	can
is	weak	is	rate
is	but	is	how
.	.	.	.
strong	are	strong	many

The number of negative examples to sample per positive example is a modeling choice.

# Parametrization of the Prediction Problems

shallow nn

$K$  is the embedding dimension, which is lower than the dimension of  $V$

Endow each word  $v$  in the vocabulary with an embedding vector  $\rho_v \in \mathbb{R}^K$  and a context vector  $\alpha_v \in \mathbb{R}^K$  where  $K \ll V$ .

The positive examples are modeled as

$$\Pr[w_{d,n-l} \in C(w_{d,n}) \mid w_{d,n}] = \frac{\exp(\rho_{w_{d,n}}^T \alpha_{w_{d,n-l}})}{1 + \exp(\rho_{w_{d,n}}^T \alpha_{w_{d,n-l}})}$$

and the negative examples are modeled as

$$\Pr[w_{d,n-l} \notin C(w_{d,n}) \mid w_{d,n}] = 1 - \frac{\exp(\rho_{w_{d,n}}^T \alpha_{w_{d,n-l}})}{1 + \exp(\rho_{w_{d,n}}^T \alpha_{w_{d,n-l}})}$$

## Example

The first row of the table above would contribute the following elements to the loss function:

$$\Pr[\text{growth} \in C(w_{d,n}) \mid w_{d,n} = \text{economic}] = \frac{\exp(\boldsymbol{\rho}_{\text{economic}}^T \boldsymbol{\alpha}_{\text{growth}})}{1 + \exp(\boldsymbol{\rho}_{\text{economic}}^T \boldsymbol{\alpha}_{\text{growth}})}$$

$$\Pr[\text{down} \notin C(w_{d,n}) \mid w_{d,n} = \text{economic}] = \frac{1}{1 + \exp(\boldsymbol{\rho}_{\text{economic}}^T \boldsymbol{\alpha}_{\text{down}})}$$

Loss function multiplies all such probabilities together and optimizes using gradient methods.

# How Close are Words?

The standard way of judging whether word vectors are “close” is **cosine similarity**.

$$CS(i, j) = \frac{\rho_i \cdot \rho_j}{\|\rho_i\| \|\rho_j\|}$$

Measures whether vectors point in the same direction.

# Terms Close to Uncertainty in FOMC Transcripts

vector representation of uncertainty  $\rho_{\text{hat\_uncertainty}}$  in  $R^k$

then list the words and the CS of other words with uncertainty

this is a data-driven way to find synonyms of uncertainty

term	sim	term	sim
uncertainties	0.741	challenges	0.415
anxiety	0.48	fragility	0.405
pessimism	0.479	clarity	0.401
skepticism	0.465	concerns	0.4
optimism	0.445	risks	0.397
caution	0.442	disagreement	0.387
gloom	0.437	volatility	0.384
uncertain	0.433	tension	0.383
sensitivity	0.427	certainty	0.382
angst	0.426	skepticism	0.38

# Terms Close to Risk

term	sim	term	sim
risks	0.737	misdirected	0.385
threat	0.609	odds	0.379
danger	0.541	uncertainty	0.375
dangers	0.463	concern	0.371
vulnerability	0.457	prospect	0.37
chances	0.451	instability	0.363
breakout	0.433	potentially	0.352
probability	0.426	concerns	0.352
possibility	0.409	challenges	0.346
likelihood	0.406	risking	0.342

# Document Similarity

Typical way of representing documents given a word embedding model is the average embedding for all words in document:

$$\rho_d = \frac{1}{N_d} \sum_n \rho_{w_{d,n}}$$

These representations can then be used to compute cosine similarity:

1. [Hansen et al., 2021] measures skill content of executive job postings.
2. [Gennaro and Ash, 2022] measures tone of political speeches.
3. [Kogan et al., 2023] measures occupational exposure to technical change. **embedding onet task and embedding patent, find cosine similarity**

# Document Similarity

Typical way of representing documents given a word embedding model is the average embedding for all words in document:

$$\rho_d = \frac{1}{N_d} \sum_n \rho_{w_{d,n}}$$

These representations can then be used to compute cosine similarity:

1. [Hansen et al., 2021] measures skill content of executive job postings.
2. [Gennaro and Ash, 2022] measures tone of political speeches.
3. [Kogan et al., 2023] measures occupational exposure to technical change.

**NB:**  $\rho_d$  is not sensitive to word order!

# Other Contextual Data

Similar algorithms can be used to represent any data where context informs relatedness.

with susan athey

[Ruiz et al., 2020] use embeddings to represent products: two products are similar when they appear in similar baskets.  
shopper model

Analogous idea for supply chains: firms are similar when they share similar co-suppliers.

# Outline

Introduction

Basic Concepts

Word Embeddings

Attention and Transformer Models

Economic Applications

# Word Prediction in Large Language Models

To simplify notation, consider sequence of words  $\mathbf{w} = (w_1, \dots, w_N)$ .

The prediction target of **autoregressive** or **generative** language models (e.g., GPT family) is  $w_N \mid \mathbf{w}_{-N}$ . **know  $w_1, w_2, \dots, w_{\{N-1\}}$ , predict  $w_N$**

In **bidirectional** models (e.g., BERT), the prediction target is  $w_n \mid \mathbf{w}_{-n}$ .

**BERT predicts a missing or masked word giving both left and right context**

In both cases, the prediction is informed by surrounding context.

**know  $w_1, w_2, \dots, w_{\{n-1\}}, w_{\{n+1\}}, \dots, w_N$ , predict  $w_n$**

# Example of Next-Word Prediction Problem

*After a season of positive corporate earnings announcements driven by AI adoption, NVIDIA's share price hit an all-time [MASK].*

Which words are most likely to underlie [MASK]?

V different word possible

full probability of all the V words

## Two Alternative Examples with Six Words Removed

*After a season of ~~positive~~ corporate earnings announcements driven by AI adoption, ~~NVIDIA's share price~~ hit an all-time [MASK].*

*After a season of positive corporate earnings announcements driven by AI ~~adoption~~, NVIDIA's share price hit an all-time [MASK].*

only certain part of the context is very relevant

# Formalizing the Prediction Problem

$\rho[\text{mask}]$  lives in  $\mathbb{R}^k$

feed forward nn to get  $\rho_{\text{hat}_v}$

LLM gives a computation-efficient way to give  $\rho[\text{mask}]$

Endow the masked word  $n$  with an embedding vector  $\rho_n \in \mathbb{R}^K$ .

$\rho_n$  can be used to fit a probability distribution over the  $V$  vocabulary terms that can populate the  $n$ th element of the sequence.

Multinomial regression / feedforward neural network with  $\rho_n$  as input.

How can  $\rho_n$  be built to reflect relevant part of the context?

Also important for construction to be computationally efficient.

# Attention Weights self attention operator

My name is Stephan

"attended word": the word one focuses on

"context word": the surrounding word

What parts of context words is helpful in understanding meaning of the attended word

The basic idea of attention [Vaswani et al., 2017] is to define a weight  $\alpha_{n,m}$  for each **attended** word  $n$  and **context** word  $m$ .

Normalized so that  $\sum_m \alpha_{n,m} = 1$ .

Attention weights highlight the relevant parts of the context surrounding each word.

Weights are estimated during neural network training to optimize the quality of word prediction tasks.

# Parameterization of Attention

Let  $\mathbf{W}_q \in \mathbb{R}^{R \times K}$  and  $\mathbf{W}_k \in \mathbb{R}^{R \times K}$  be **query** and **key** weight matrices.

Steps to generate attention weight  $\alpha_{n,m}$ :

1. Form query vector  $\mathbf{q}_n = \mathbf{W}_q \boldsymbol{\rho}_n$
2. Form key vector  $\mathbf{k}_m = \mathbf{W}_k \boldsymbol{\rho}_m$
3. Compute score  $\tilde{\alpha}_{n,m} = \frac{\mathbf{q}_n \cdot \mathbf{k}_m}{\sqrt{R}}$  **dot product**
4.  $\alpha_{n,m} = \frac{\exp(\tilde{\alpha}_{n,m})}{\sum_{m'} \exp(\tilde{\alpha}_{n,m'})}$

**GPU can do quick matrix multiplication**

**the goal is still to maximizing predicting of presence or absence of word**

# Using Attention to Update Embeddings

Suppose we have an initial embedding representation  $\rho_n^{(i)}$  for word  $n$ .

Let  $\mathbf{W}_v \in \mathbb{R}^{S \times K}$  be matrix of **value** weights.

Project embedding into value vector  $\mathbf{v}_n^{(i)} = \mathbf{W}_v \rho_n^{(i)}$ .

We obtain a new representation for word  $n$  via

$$\rho_n^{(i+1)} = \mathbf{W}_0 \sum_m \alpha_{n,m} \mathbf{v}_m^{(i)}$$

where  $\mathbf{W}_0 \in \mathbb{R}^{K \times S}$ .

# Transformer Model

A Transformer model is a deep learning model that repeatedly applies attention operations to initial word embeddings.

Inside a single Transformer block, attention operations are applied multiple times in parallel via [multi-head attention](#).

Each updated word embedding is then passed through a [feedforward neural network](#) to non-linearly transform it prior to entering the next Transformer block.

# Initial Embeddings

to be or not to be

the first be has  $\rho_{be}^{(0)} + \rho_2^{(0)}$

the second be has  $\rho_{be}^{(0)} + \rho_6^{(0)}$

Each word in the input sequence has an initial embedding vector that is the sum of two distinct embeddings:

1. An **embedding for the vocabulary term**.
2. A **positional embedding** that depends on the location of the word in the sequence.

These embeddings are additional estimated network parameters.

Given the estimated structure of the whole network, every input sequence can be processed even if it was not seen in training data.

# Summary of Structure of Large Language Model

1. Begin with input sequence  $w_1, \dots, w_N$ .
2. Assign initial embeddings to each element of sequence.
3. Repeatedly perform the following operations:
  - 3.1 Linearly combine embeddings with attention weights.
  - 3.2 Non-linearly transform each embedding with feed-forward neural network.
4. Output final embeddings for each element of sequence.
5. Use final embeddings for language prediction problem.

# Modeling Choices

While this basic pipeline describes nearly every LLM, there is variety in:

1. Prediction target (e.g. bidirectional vs. autoregressive)
2. Training data
3. Length of context window
4. Number of Transformer blocks
5. Dimensionality of embedding vectors

# BERT

this is open sourced  
it may be good it use a fine-tuned bert, because it is more transparent  
and prelicable

Important example is BERT (Bidirectional Encoding Representations from Transformers) [Devlin et al., 2019].

Trained on BooksCorpus (800M words) and English Wikipedia (2,500M words).

**Masked language modeling.** 15% of words randomly masked and given [MASK] token. [MASK] token embeddings built to successfully predict underlying word.

Original paper had next-sentence prediction but has since been dropped from loss function in extensions [Liu et al., 2019].

Base model has twelve layers, 768-dimensional embeddings, 110M parameters.

# Which Corpus?

Much of traditional text-as-data analysis fits models on corpora drawn from domain of interest.

Large language models were first fit on generic corpora like Common Crawl, Wikipedia, or Google Books.

More recent iterations expand the training data (but details becoming more obscure).

Important to realize that **the training data contains the knowledge that a model can encode.**

Any biases in the training data can also be inherited by the model.

# Example GPT-3 Output

GPT-4 has post training to fix this

Prompt GPT-3 [Brown et al., 2020] with He was very [MASK] and She was very [MASK].

**Table 6.1:** Most Biased Descriptive Words in 175B Model

Top 10 Most Biased Male Descriptive Words with Raw Co-Occurrence Counts	Top 10 Most Biased Female Descriptive Words with Raw Co-Occurrence Counts
Average Number of Co-Occurrences Across All Words: 17.5	Average Number of Co-Occurrences Across All Words: 23.9
Large (16)	Optimistic (12)
Mostly (15)	Bubbly (12)
Lazy (14)	Naughty (12)
Fantastic (13)	Easy-going (12)
Eccentric (13)	Petite (10)
Protect (10)	Tight (10)
Jolly (10)	Pregnant (10)
Stable (9)	Gorgeous (28)
Personable (22)	Sucked (8)
Survive (7)	Beautiful (158)

# Outline

Introduction

Basic Concepts

Word Embeddings

Attention and Transformer Models

Economic Applications

# Document Similarity

LLMs produce vector representations of word sequences which can be used in place of non-contextual methods.

Plausible argument for preferring bidirectional models, which exploit full structure of document.

LLMs can produce different vectors for

*economic growth is weak but long-term productivity trends are strong*  
*economic growth is strong but long-term productivity trends are weak.*

# Document Similarity

LLMs produce vector representations of word sequences which can be used in place of non-contextual methods.

Plausible argument for preferring bidirectional models, which exploit full structure of document.

LLMs can produce different vectors for

*economic growth is weak but long-term productivity trends are strong*  
*economic growth is strong but long-term productivity trends are weak.*

**Open question** is to what extent this improves similarity measures.

# Zero/Few-Shot Learning

The structure of autoregressive language models suggests their use for directly measuring  $\theta_i$ .

# Zero/Few-Shot Learning

The structure of autoregressive language models suggests their use for directly measuring  $\theta_i$ .

Formulate a sequence of words such as

*“Consider the sentence ‘the economy is booming’. Is the sentiment in this sentence positive, neutral, or negative?”*

This text is converted to  $\mathbf{w} = (w_1, \dots, w_N)$ .

$N + 1$ th word is drawn from  $\Pr[w_{N+1} \mid \mathbf{w}]$ .

$N + 2$ th word is drawn from  $\Pr[w_{N+2} \mid \mathbf{w}, w_{N+1}]$ , and so forth.

Already by GPT-2 emergent [zero-shot learning](#) behavior observed from training on next-word prediction loss.

# Zero/Few-Shot Learning

The structure of autoregressive language models suggests their use for directly measuring  $\theta_i$ .

Formulate a sequence of words such as

*“Consider the sentence ‘the economy is booming’. Is the sentiment in this sentence positive, neutral, or negative?”*

This text is converted to  $\mathbf{w} = (w_1, \dots, w_N)$ .

$N + 1$ th word is drawn from  $\Pr[w_{N+1} \mid \mathbf{w}]$ .

$N + 2$ th word is drawn from  $\Pr[w_{N+2} \mid \mathbf{w}, w_{N+1}]$ , and so forth.

Already by GPT-2 emergent [zero-shot learning](#) behavior observed from training on next-word prediction loss.

See [Bybee, 2023] and [Hansen and Kazinnik, 2024] for early applications in economics.

# Generating Structured Information

Even outside economic measurement, LLMs add value by mapping unstructured data into structured representations.

For example, extracting specific narratives from text, e.g. “who did what to whom.”

LLMs largely overcome need for OCR conversion of images to text.

Very useful for fields like economic history.

# Conclusion

Foundation LLMs are large-scale word prediction machines.

But word prediction machines can be surprisingly useful!

Lack of benchmarking on economic tasks makes quantifying value added to measurement a challenge.

For any given gain in measurement accuracy, need to trade off lack of transparency and reproducibility.

Deeper questions surrounding LLMs ability to learn realistic “world model” [Vafa et al., 2024].

As with word2vec, more general idea of embedding sequential data [Gabaix et al., 2023].

# References I

Adukia, A., Eble, A., Harrison, E., Runesha, H. B., and Szasz, T. (2023).

What We Teach About Race and Gender: Representation in Images and Text of Children's Books.

[The Quarterly Journal of Economics](#), 138(4):2225–2285.

Ash, E. and Hansen, S. (2023).

Text Algorithms in Economics.

[Annual Review of Economics](#), 15(1):659–688.

Ash, E., Hansen, S., Marangon, C., and Muvdi, Y. (2025).

Large language models in economics.

In [Handbook of Economics and Language](#), Palgrave Handbooks.

Baker, S. R., Bloom, N., and Davis, S. J. (2016).

Measuring Economic Policy Uncertainty.

[The Quarterly Journal of Economics](#), 131(4):1593–1636.

Bandiera, O., Prat, A., Hansen, S., and Sadun, R. (2020).

CEO Behavior and Firm Performance.

[Journal of Political Economy](#), 128(4):1325–1369.

# References II

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003).

A neural probabilistic language model.

[The Journal of Machine Learning Research](#), 3(null):1137–1155.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020).

Language Models are Few-Shot Learners.

Bybee, J. L. (2023).

The Ghost in the Machine: Generating Beliefs with Large Language Models.

Working Paper.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019).

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

In [Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 \(Long and Short Papers\)](#), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

# References III

Gabaix, X., Koijen, R. S. J., and Yogo, M. (2023).

Asset Embeddings.

[SSRN Electronic Journal](#).

Gennaro, G. and Ash, E. (2022).

Emotion and Reason in Political Language.

[The Economic Journal](#), 132(643):1037–1059.

Gentzkow, M., Kelly, B., and Taddy, M. (2019a).

Text as Data.

[Journal of Economic Literature](#), 57(3):535–574.

Gentzkow, M., Shapiro, J. M., and Taddy, M. (2019b).

Measuring Group Differences in High-Dimensional Choices: Method and Application to Congressional Speech.

[Econometrica](#), 87(4):1307–1340.

Gorodnichenko, Y., Pham, T., and Talavera, O. (2023).

The Voice of Monetary Policy.

[American Economic Review](#), 113(2):548–584.

# References IV

Hansen, A. L. and Kazinnik, S. (2024).

Can ChatGPT Decipher Fedspeak?

Hansen, S., Ramdas, T., Sadun, R., and Fuller, J. (2021).

The Demand for Executive Skills.

Technical Report 28959, National Bureau of Economic Research, Inc.

Jurafsky, D. and Martin, J. H. (2025).

Speech and Language Processing.

3rd edition.

Kogan, L., Papanikolaou, D., Schmidt, L. D., and Seegmiller, B. (2023).

Technology and Labor Displacement: Evidence from Linking Patents with Worker-Level Data.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019).

RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Ludwig, J., Mullainathan, S., and Rambachan, A. (2025).

Large Language Models: An Applied Econometric Framework.

# References V

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a).

Efficient Estimation of Word Representations in Vector Space.

[arXiv:1301.3781 \[cs\]](#).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b).

Distributed Representations of Words and Phrases and their Compositionality.

[arXiv:1310.4546 \[cs, stat\]](#).

Plaza-del-Arco, F. M., Nozza, D., and Hovy, D. (2024).

Wisdom of Instruction-Tuned Language Model Crowds. Exploring Model Label Variation.

In Abercrombie, G., Basile, V., Bernadi, D., Dudy, S., Frenda, S., Havens, L., and Tonelli, S., editors, [Proceedings of the 3rd Workshop on Perspectivist Approaches to NLP \(NLPerspectives\) @ LREC-COLING 2024](#), pages 19–30, Torino, Italia. ELRA and ICCL.

Ruiz, F. J. R., Athey, S., and Blei, D. M. (2020).

SHOPPER: A probabilistic model of consumer choice with substitutes and complements.

[The Annals of Applied Statistics](#), 14(1):1–27.

# References VI

Vafa, K., Chen, J. Y., Kleinberg, J., Mullainathan, S., and Rambachan, A. (2024).  
Evaluating the World Model Implicit in a Generative Model.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017).

Attention is All you Need.

In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc.