Unstructured Data for Economics

Lecture 2:
Large Language Models

Stephen Hansen
University College London

Reading

Background material for the lecture: Jurafsky and Martin, *Speech and Language Processing*, Ch. 10 `https://bityl.co/QQRX`.

Limitations of Bag-of-Words + Cosine Similarity

Synonomy

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 + Cosine Similarity

Synonomy

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 + Cosine Similarity

Synonomy

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

Sequence

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

Outline

1. Attention and Word Prediction
2. Applications

Word Prediction and Attention

Word Prediction in Large Language Models

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

The prediction target of autoregressive or generative language models (e.g., GPT family) is $w_N \mid \mathbf{w}_{-N}$.

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

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

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]?

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].

Formalizing the Prediction Problem

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

$\boldsymbol{\rho}_n$ can used to fit a probability distribution over the V vocabulary terms that can populate the nth element of the sequence.

Multinomial regression / feedforward neural network with $\boldsymbol{\rho}_n$ as input.

How can $\boldsymbol{\rho}_n$ be built to reflect relevant part of the context?

Also important for construction to be computationally efficient.

Attention Weights

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 $\tilde{\alpha}_{n,m} = \frac{\mathbf{q}_n \cdot \mathbf{k}_m}{\sqrt{R}}$
4. $\alpha_{n,m} = \frac{\exp(\tilde{\alpha}_{n,m})}{\sum_{m'} \exp(\tilde{\alpha}_{n,m'})}$

Using Attention to Update Embeddings

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

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

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

We obtain a new representation for word n via

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

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

Attention is Matrix Multiplication

Suppose we stack all the embeddings in a $K \times N$ matrix $\mathbf{P}^{(i)}$.

Query vectors can be obtained through $\mathbf{Q} = \mathbf{W}_q \mathbf{P}^{(i)} \in \mathbb{R}^{R \times N}$.

Key vectors can be obtained through $\mathbf{K} = \mathbf{W}_k \mathbf{P}^{(i)} \in \mathbb{R}^{R \times N}$.

Unnormalized scores are $\tilde{\mathbf{A}} = \mathbf{Q}^T \mathbf{K} \in \mathbb{R}^{N \times N} \longrightarrow$ attention weights $\mathbf{A}$ obtained by row normalization with multinomial logistic.

Value vectors can be obtained through $\mathbf{V} = \mathbf{W}_v \mathbf{P}^{(i)} \in \mathbb{R}^{S \times N}$.

Final update is $P^{(i+1)} = \mathbf{W}_0 \mathbf{V} \mathbf{A}^T$.

No need to sequentially pass through the data.

Multi-Head Attention

Words may be in relationship in many ways.

Different attention weights are typically applied in parallel to update initial embeddings.

The full operation is called multi-head attention.

Each attention operation $h = 1, \ldots, H$ has its own weight matrices.

Fully updated embedding vector is

$$\boldsymbol{\rho}_n^{(i+1)} = \sum_{h=1}^{H} \mathbf{W}_{0,h} \sum_{m=1}^{N} \alpha_{n,m}^{h} \mathbf{v}_m^{(i),h}$$

Full Transformer Block

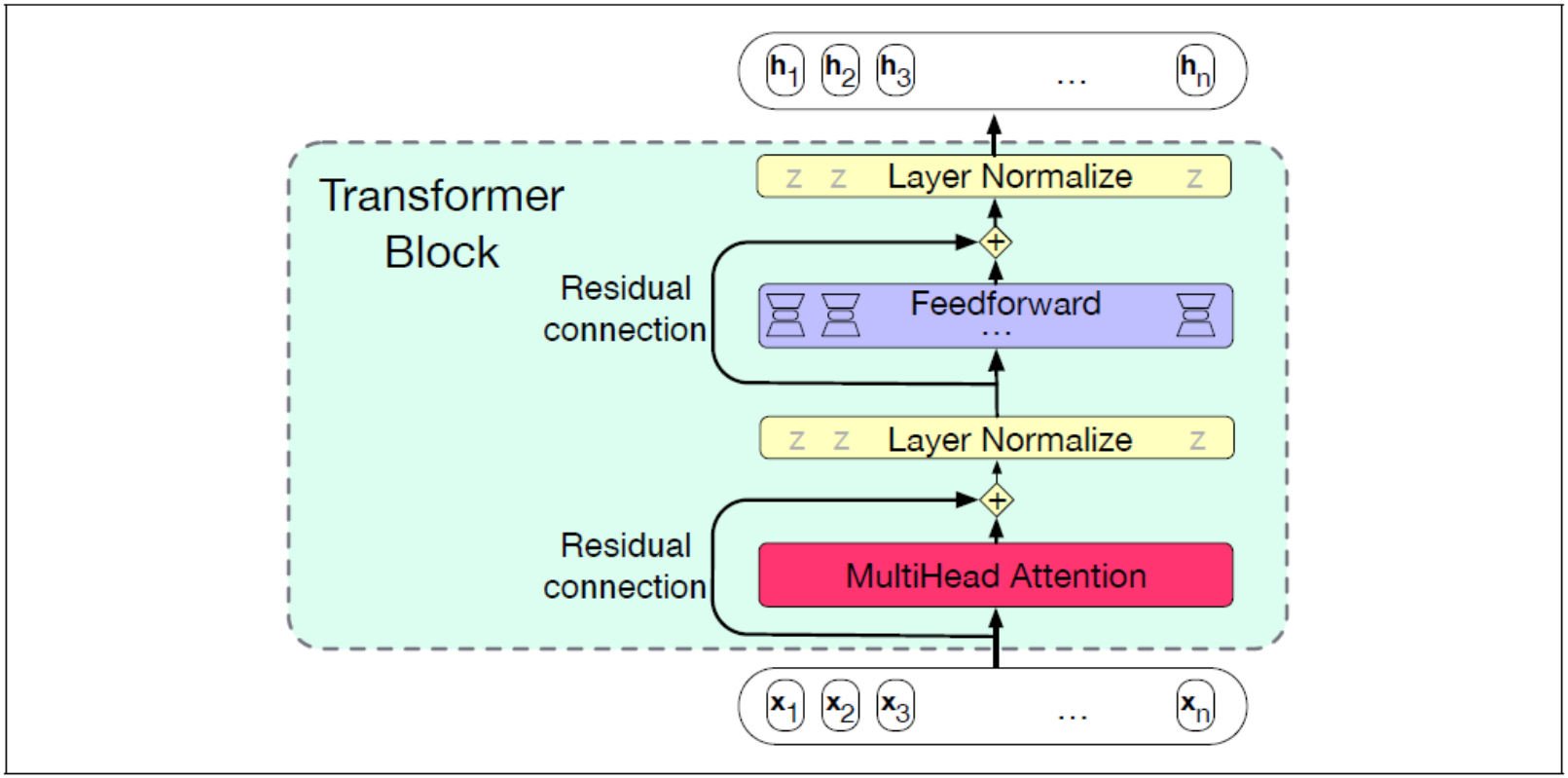


Figure 10.6 A transformer block showing all the layers.

Initial Embeddings

Transformer blocks take as inputs a sequence of embeddings and output an updated sequence of embeddings.

But so far we have not specified where the initial embeddings come from.

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 processed even if it was not seen in training data.

Summary of Structure of Large Language Model

1. Begin with input sequence $w_1, \ldots, w_N$.
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

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.

Model Fitting

Small-scale Transformer models can be built and fit locally.

See `https://www.youtube.com/watch?v=l8pRSuU81PU` for end-to-end implementation of GPT-2.

In most cases, fitting LLMs is sufficiently capital intensive that only large organizations do so.

Individual users (i) download the fitted model to use or repurpose; or, (ii) interact with the fitted model via an owner-provided interface.

Applications

Distance between Documents

Both bidirectional and autoregressive language models produce vector representations of word sequences.

Cosine similarity between these representations can then be used to measure distance.

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

Unlike previous methods, 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

Measuring Sentiment Towards Finance

[Jha et al., 2022] uses BERT to measure how the financial sector is discussed in general language.

Define positive and negative sentiment sentences and obtain their BERT embeddings.

The difference in embeddings defines a "net sentiment" direction in the vector space.

Any other sentence can be projected onto this direction using cosine similarity.

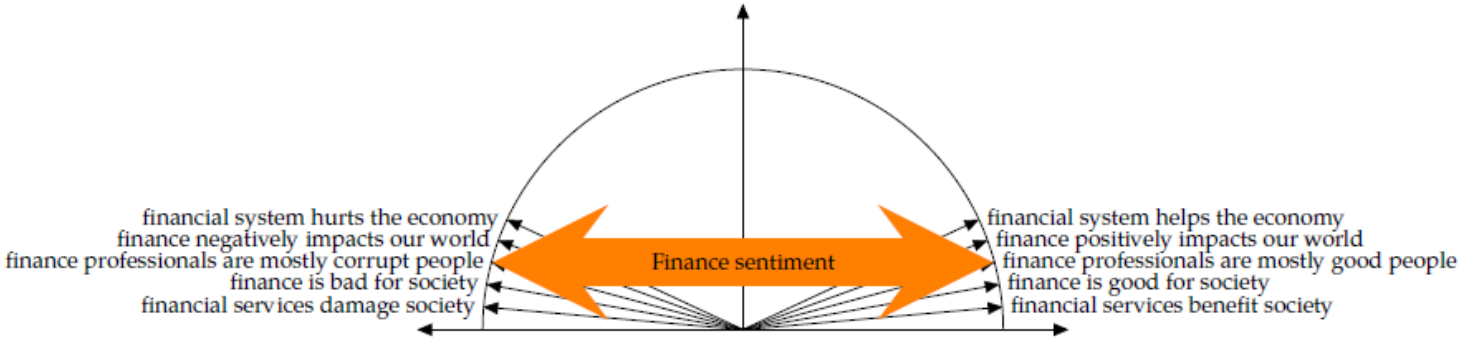
Builds on work of [Kozlowski et al., 2019] which performs similar exercise using word2vec.

Table 2: Positive − negative defining sentences for English

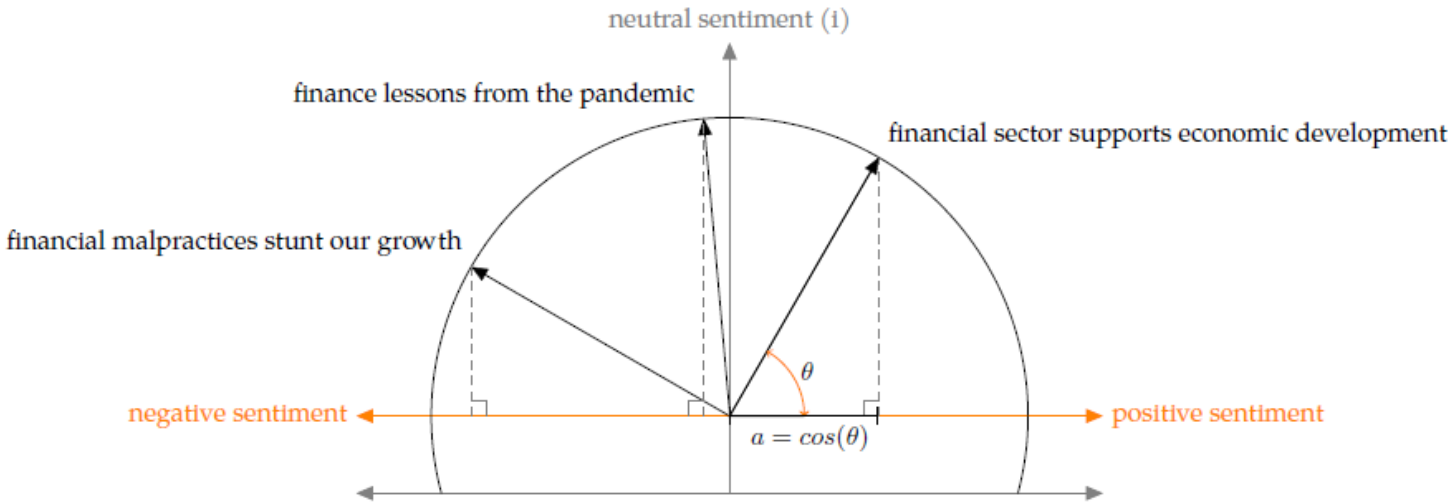
Positive sentences	Negative sentences
financial services benefit society	financial services damage society
finance is good for society	finance is bad for society
finance professionals are mostly good people	finance professionals are mostly corrupt people
finance positively impacts our world	finance negatively impacts our world
financial system helps the economy	financial system hurts the economy

Note: To define the positive minus negative dimension, we average the embeddings of positive sentences less that of their negative counterparts.

Figure 1: Conceptual diagram of finance sentiment measurement



(a) Defining the positive minus negative finance sentiment dimension



(b) Projection of sentences onto positive minus negative sentiment dimension

Table 3: Sentences assigned the most positive and negative finance sentiment for American English

Positive sentiment sentences	Negative sentiment sentences
the goal of financial management	turmoil in the financial markets
finance in the graduate school	finances become disordered the
financial support of the center	financial panic swept the country
financial management of the organization	turmoil in financial markets
business and financial experience	financial panic swept the nation
financial support of the graduate	instability in the financial markets
financial support of the science	financial panic in the country
financial support of the course	severe financial setbacks
financial support of the field	a major financial panic
knowledge of the financial structure	world wide financial panic

Note: A sentence is assigned positive or negative finance sentiment, based on its projection onto the finance positivity dimension (cosine similarity). Sentences at the top are the most positive or negative in their respective column, and the absolute value of finance sentiment decreases down each list.

Concept Detection

The structure of autoregressive language models suggests their use for concept detection.

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 $\mathbf{w} = (w_1, \ldots, w_N)$.

$N + 1$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.

The ability to generate seemingly correct responses is called zero-shot learning.

Temperature

One important modeling choice is how exactly to sample from $\Pr[\, w_{N+1} \mid \mathbf{w} \,]$.

Always choosing most likely word can make response rigid.

Sampling from the full distribution allows more creativity but can produce less accurate responses.

The temperature is a parameter that controls the trade-off between quality and diversity.

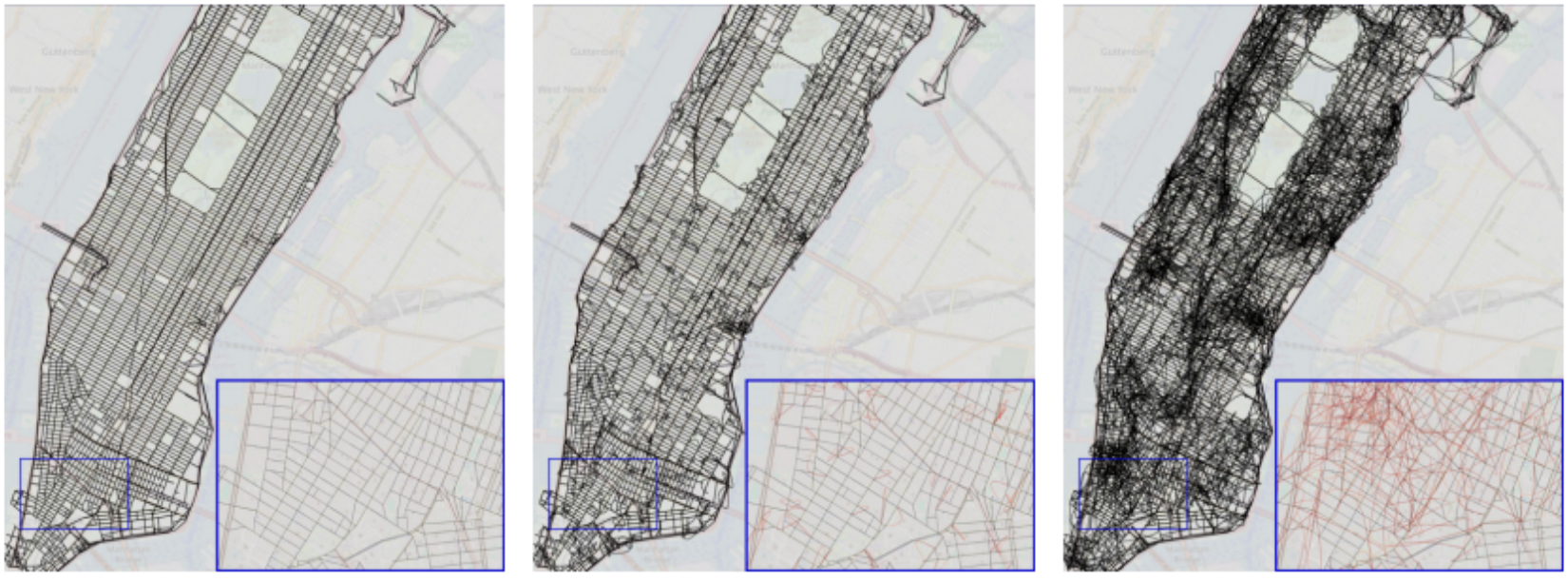
Example

A large literature seeks to classify central bank documents as having a hawkish or dovish tone.

This is a concept detection problem for which LLMs can be deployed.

See [Hansen and Kazinnik, 2024] for example.

Accurate Prediction $\neq$ True Model



(a) World model **(b)** World model with noise **(c)** Transformer

Figure 3: Reconstructed maps of Manhattan from sequences produced by three models: the true world model (left), the true world model corrupted with noise (middle), and a transformer trained on random walks (right). Edges exit nodes in their specified cardinal direction. In the zoomed-in images, edges belonging to the true graph are black and false edges added by the reconstruction algorithm are red. We host interactive reconstructed maps from transformers at the following links: shortest paths, noisy shortest paths, and random walks.

From [Vafa et al., 2024].

References I

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.

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

Can ChatGPT Decipher Fedspeak?

Jha, M., Liu, H., and Manela, A. (2022).

Does Finance Benefit Society? A Language Embedding Approach.

Kozlowski, A. C., Taddy, M., and Evans, J. A. (2019).

The Geometry of Culture: Analyzing the Meanings of Class through Word Embeddings.

American Sociological Review, 84(5):905–949.

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.

References II

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.