# Discussion:

# Cognitive Embeddings

Laurenz De Rosa

Yinan Su
*Johns Hopkins University Carey Business School*

Yiran Fan Memorial Conference
May 22, 2026

# Summary

- "Cognitive Embeddings"
  to recover analysts' cognitive state proxies from text (reports) and observed behavior (earnings forecasts)

# Summary

- "Cognitive Embeddings"
  to recover analysts' cognitive state proxies from text (reports) and observed behavior (earnings forecasts)
- Setting:
  - earnings calls $\rightarrow$ analyst reports $\rightarrow$ earnings forecasts

# Summary

- "Cognitive Embeddings"
  to recover analysts' cognitive state proxies from text (reports) and observed behavior (earnings forecasts)
- Setting:
  - earnings calls $\rightarrow$ analyst reports $\rightarrow$ earnings forecasts
- Implementation:
  - embedding = textual-attention-model(call, report)
    statistically predicts
    analyst forecasts and forecast errors

# Summary

- "Cognitive Embeddings"
  to recover analysts' cognitive state proxies from text (reports) and observed behavior (earnings forecasts)
- Setting:
  - earnings calls $\rightarrow$ analyst reports $\rightarrow$ earnings forecasts
- Implementation:
  - embedding = textual-attention-model(call, report)
    statistically predicts
    analyst forecasts and forecast errors
- To claim success:
  - the prediction is accurate
  - interpretable predictive patterns

# The model

- Cognitive embedding = Textual-attention-model(call, report)

# Textual embedding

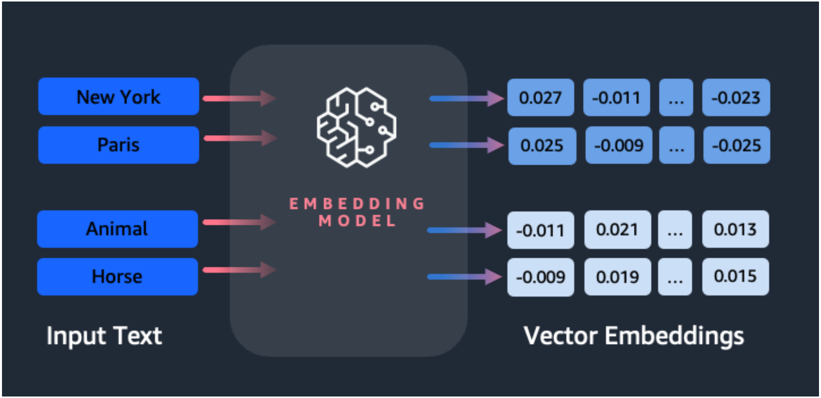



illustration source: AWS Machine Learning Blog.

https://aws.amazon.com/blogs/machine-learning/getting-started-with-amazon-titan-text-embeddings/

# Semantic similarity

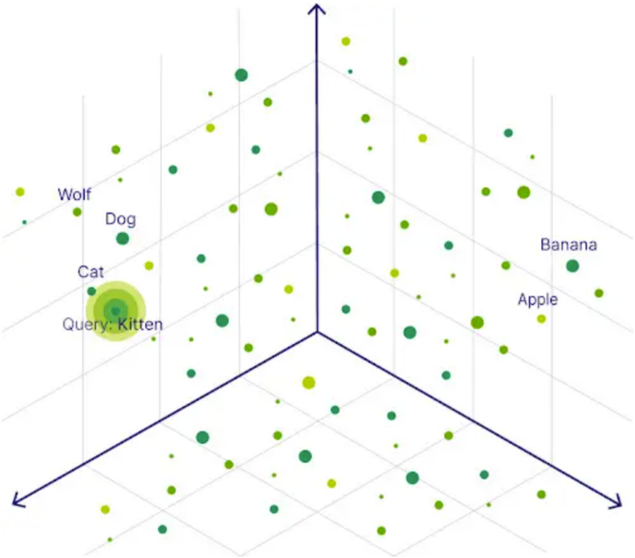



illustration source: From prototype to production: Vector databases in generative AI applications.
https://stackoverflow.blog/2023/10/09/from-prototype-to-production-vector-databases-in-generative-ai-applications/

# The model

- Cognitive embedding = Textual-attention-model(call, report)

# The model

- Cognitive embedding = Textual-attention-model(call, report)
- My intuitive interpretation:
- *Analyst highlighted version of call text*
- *The cognitive embedding is a report attention-weighted average of earnings call snippet embeddings*
- The attention mechanism:
  - Call is chopped into $N_k(t)$ snippets, each with an embedding $k_{jt}$.
  - Report assigns attention weights $w_{jt}$ to each snippet, based on the similarity between the call-report snippets
  - Then re-weight the call's snippet embeddings

$$e_{t,h} = \sum_{j=1}^{N_k(t)} w_{jt,h} k_{jt}$$

# The model

- Cognitive embedding = Textual-attention-model(call, report)
- My intuitive interpretation:
  - *Analyst highlighted version of call text*
  - *The cognitive embedding is a report attention-weighted average of earnings call snippet embeddings*
- The attention mechanism:
  - Call is chopped into $N_k(t)$ snippets, each with an embedding $k_{jt}$.
  - Report assigns attention weights $w_{jt}$ to each snippet, based on the similarity between the call-report snippets
  - Then re-weight the call's snippet embeddings

$$e_{t,h} = \sum_{j=1}^{N_k(t)} w_{jt,h} k_{jt}$$

# The model

- Cognitive embedding = Textual-attention-model(call, report)
- My intuitive interpretation:
  - *Analyst highlighted version of call text*
  - *The cognitive embedding is a report attention-weighted average of earnings call snippet embeddings*
- The attention mechanism:
  - Call is chopped into $N_k(t)$ snippets, each with an embedding $k_{jt}$.
  - Report assigns attention weights $w_{jt}$ to each snippet, based on the similarity between the call-report snippets
  - Then re-weight the call's snippet embeddings

$$e_{t,h} = \sum_{j=1}^{N_k(t)} w_{jt,h} k_{jt}$$

- Connecting economics/behavior models to machine learning models

# Comments

- How to go from the results to the claim of cognitive state?
- What should **ideal cognitive state** look like?
  - ideal information source and data structure
  - ideal predictive performance

# Comments

Ideal cognitive state?

- It should be a property of the analyst, after receiving the call information
- Data structure: one cognitive state per analyst-call pair?
  - one single embedding that covers different aspects: across forecast horizons, different forecast objects, etc.
  - the paper has different embeddings for different forecast horizons, and for forecast errors vs. levels $e^V_{t,h}, e^E_{t,h}$.
  - in terms of architecture, consider using a shared embedding across forecast objects and have different downstream prediction heads
  - more in line with the idea of a single cognitive state relevant for different purposes

# Comments

Ideal cognitive state?

- Sequential persistence and updates
  - Each call-report pair independently processed?

1. How about $e = model(\text{this report}, \text{last report})$?
2. Or even an RNN architecture?

   $e_{\text{this report}} = model(e_{\text{last report}}, \text{this report})$

   a memory component, so that the cognitive state is updated sequentially over a new call-report pair
3. Coibion-Gorodnichenko regressions:
   - forecast errors on **forecast revisions**
   - cf. "Analysts' Belief Formation in Their Own Words" (Ke 2025): CG regression instrumented by topic-level textual embeddings from analysts' own report language.

# Prediction accuracy evaluation ($R^2$)

- Main evaluation metric:
  "*Analyst-disjoint*" out-of-sample $R^2$
- Main baseline:
  "*query-shuffle falsification*"
  To represent: "merely extracting generic call-level predictability"

# Prediction accuracy evaluation ($R^2$)

- Main evaluation metric:
  "*Analyst-disjoint*" out-of-sample $R^2$
- Main baseline:
  "*query-shuffle falsification*"
  To represent: "merely extracting generic call-level predictability"
- My suggestion for baseline:
  Call fixed effects
  call-level average forecast as a predictor

# Predicting forecast levels vs. forecast errors

- Forecast levels and errors are mechanically related via realized EPS.

$$E_{a,c} = V_{a,c} - \mathsf{EPS}_c$$

- $\mathsf{EPS}_c$ is a call-level fixed effect, same across analysts (no $a$ subscript)
- Predicting errors = Predicting levels + Predicting call-fixed true EPS (there should be a formal decomposition of MSE and $R^2$)

# Predicting forecast levels vs. forecast errors

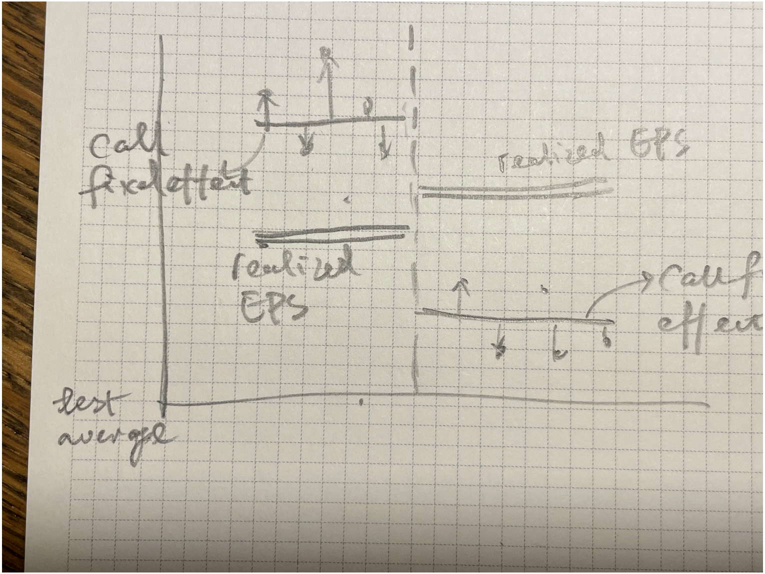
- Forecast levels and errors are mechanically related via realized EPS.

$$E_{a,c} = V_{a,c} - \mathsf{EPS}_c$$

- $\mathsf{EPS}_c$ is a call-level fixed effect, same across analysts (no $a$ subscript)
- Predicting errors = Predicting levels + Predicting call-fixed true EPS
  (there should be a formal decomposition of MSE and $R^2$)
- The difference is predicting future EPS from the call.

  That may be economically valuable, but it is not necessarily related to analyst cognition.

# Forecast errors and call fixed effects



# Other prediction comments

1. Do we worry about direct information revelation from the report about forecasts?
   - Directly talking about forecasts in the report
   - Or, more subtly

     Say a report always has two sections: good things and bad things. Then the report allocates biased attention to the two sections

     (is this the cognitive embedding that we want or this is leak?)
2. Are the benchmarks informative?
   - "Mean-head":

     "replaces learned attention with the mean call embedding and *keeps the value head fixed*"

     can be too harsh, because the value head is trained to work with the attention-weighted embedding, not the mean embedding

# What else can we do with ideal cognitive state?

- Predict “behavior” beyond forecasts
  - how will you write your next report, conditional on the next call?

# Discussion:

# Cognitive Embeddings

Laurenz De Rosa

Yinan Su
*Johns Hopkins University Carey Business School*

Yiran Fan Memorial Conference
May 22, 2026