Asset Embeddings

Xavier Gabaix Ralph Koijen Robert Richmond Motohiro Yogo

Harvard - Chicago - NYU - Princeton

May 2024

Identifying similar firms

- In economics, we often try to find similar firms or assets.
  - E.g., in terms of growth rates, expected returns, risk, asset substitution, product markets, ...
- Common practice: Use observable characteristics.
  - E.g., industry definitions, accounting data, ...

Identifying similar firms

- In economics, we often try to find similar firms or assets.
  - E.g., in terms of growth rates, expected returns, risk, asset substitution, product markets, ...
- Common practice: Use observable characteristics.
  - E.g., industry definitions, accounting data, ...
- Those characteristics may be quite imperfect.
  - Standardized accounting data are an incomplete summary.
    - E.g., number of subscribers at Netflix, ...
  - New economic environments call for creative, new characteristics.
    - E.g., exposure to COVID-19, growth in intangibles, ...

Identifying similar firms

- In economics, we often try to find similar firms or assets.
  - E.g., in terms of growth rates, expected returns, risk, asset substitution, product markets, ...
- Common practice: Use observable characteristics.
  - E.g., industry definitions, accounting data, ...
- Those characteristics may be quite imperfect.
  - Standardized accounting data are an incomplete summary.
    - E.g., number of subscribers at Netflix, ...
  - New economic environments call for creative, new characteristics.
    - E.g., exposure to COVID-19, growth in intangibles, ...
- This paper: Use asset embeddings to measure firm similarity.

What are embeddings?

- Embeddings: Represent data (e.g., words) as continuous vectors in a potentially high-dimensional space: $x_a \in \mathbb{R}^N$.
- Embeddings play a central role in the development of large language models.
- In NLP, embeddings capture the similarity between words and it allows us to do "math with words:

$$x_{\text{Paris}} - x_{\text{France}} + x_{\text{Spain}} \simeq x_{\text{Madrid}}.$$

What are embeddings?

- Embeddings: Represent data (e.g., words) as continuous vectors in a potentially high-dimensional space: $x_a \in \mathbb{R}^N$.
- Embeddings play a central role in the development of large language models.
- In NLP, embeddings capture the similarity between words and it allows us to do “math with words:

$$x_{\text{Paris}} - x_{\text{France}} + x_{\text{Spain}} \simeq x_{\text{Madrid}}.$$

- The dense embedding vectors are learned from (lots of) data (not preselected).
- Despite the success of embedding techniques in these fields, their application in finance and economics largely unexplored.

Which data to use to learn embeddings?

- We introduce the concept of asset embeddings.
  - A vector representation per asset that we learn from data.
- Which data to use?

WHICH DATA TO USE TO LEARN EMBEDDINGS?

- We introduce the concept of asset embeddings.
  - A vector representation per asset that we learn from data.
- Which data to use?
- Our answer: Just like
  - documents organize words in NLP,
  - images organize pixels in vision,
  - songs organize notes in audio,

  investors organize assets in finance and economics.

Which data to use to learn embeddings?

- We introduce the concept of asset embeddings.
  - A vector representation per asset that we learn from data.
- Which data to use?
- Our answer: Just like
  - documents organize words in NLP,
  - images organize pixels in vision,
  - songs organize notes in audio,

  investors organize assets in finance and economics.
- Theoretically, we show how embeddings can be recovered by "inverting the asset demand system."

Which method to learn embeddings?

- Which method to use?

Which method to learn embeddings?

- Which method to use?
- Traditional approach: LSA (Latent Semantic Analysis), which is analogous to PCA/recommender systems.
- The recent ML/AI literature went way beyond that:
  - Context-invariant embeddings: E.g., GloVe and Word2Vec.
  - Embeddings with context: E.g., transformer models (e.g., BERT and GPT).
  - Parameters are estimated using masked language modeling.

Investor embeddings

- Holdings data vary by asset, investor, and time.
- Even though our focus is on asset embeddings, we obtain investor embeddings as a by-product: $\lambda_{it} \in \mathbb{R}^K$.
  - Learned vector representations of investors.

Investor embeddings

- Holdings data vary by asset, investor, and time.
- Even though our focus is on asset embeddings, we obtain investor embeddings as a by-product: $\lambda_{it} \in \mathbb{R}^K$.
  - Learned vector representations of investors.
- Potential applications:
  - Identify crowded trades.
  - Performance measurement (extending Daniel, Grinblatt, Titman, and Wermers, 1997).
  - Classify investors beyond institutional type, size, and activeness, ...
  - ...

Five main contributions

1. Uncover characteristics relevant to investors by "inverting" the asset demand system.
2. Six benchmarks to compare any type of asset embeddings.
   - Benchmarks play a key role in developing GenAI models.
3. Use various language model architectures to learn asset embeddings, including transformer models.
4. Implement the models using 13F and funds data.
   - Observed characteristics and LLM-based embeddings (Cohere and OpenAI) provide a reference point.
5. Interpretability: Use a RAG-based LLM system based on earnings calls data to interpret the learned embeddings.
   - Extends to any other form of text data (e.g., WSJ articles).

Related literature

- Demand system asset pricing.
  - Frameworks to jointly understand prices, characteristics, and holdings data.
- Machine learning and asset pricing, in particular:
  - Use (lots of) observable characteristics and price-based variables to predict future returns and risk.
  - Recent literature explores information in text data.
    - Newspapers, 10-K filings, earnings calls, social media, ...
  - See Kelly and Xiu (2023) for a recent review.
- Audio, NLP, and vision models.
  - Most closely related to embedding, transformer, and topic models.

Outline

- Inverting the asset demand system: Using holdings data as embeddings data.
- Methods to estimate embeddings.
- Data.
- Benchmarking asset embeddings.
- Empirical results.
- Extensions.

Holdings data as embeddings data

- Model the log dollar holdings of investor i in asset (i.e. stock) a as
$$h_{ia} = c_i^h + (1 - \zeta_i) p_a + \nu_{ia},$$
where ζ_i is the demand elasticity and ν_{ia} a stock-specific demand shifter.

Holdings data as embeddings data

- Model the log dollar holdings of investor i in asset (i.e. stock) a as

$$h_{ia} = c_i^h + (1 - \zeta_i)p_a + \nu_{ia},$$

where ζ_i is the demand elasticity and ν_{ia} a stock-specific demand shifter.

- We model the demand shifter as

$$\nu_{ia} = \lambda_i^{\nu\prime} x_a + u_{ia},$$

which can be micro-founded by (Koijen and Yogo, 2019):

  - Investors having mean-variance demand.
  - Returns follow a factor model.
  - Expected returns and factor loadings are affine in x_a.

Holdings data as embeddings data

- A log-linear approximation to the market clearing condition, $\sum_i \exp(h_{ia}) = \exp(p_a)$, implies:

$$p_a = c^p + \frac{1}{\zeta_S}\lambda_S^{\nu\prime} x_a + u_{Sa},$$

with $y_S \equiv \sum_i S_i^a y_i$.

Holdings data as embeddings data

- A log-linear approximation to the market clearing condition, $\sum_i \exp(h_{ia}) = \exp(p_a)$, implies:

$$p_a = c^p + \frac{1}{\zeta_S}\lambda_S^{\nu\prime} x_a + u_{Sa},$$

with $y_S \equiv \sum_i S_i^a y_i$.

- If we substitute the price back into the demand equation:

$$h_{ia} = \phi_i^h + \phi_a^h + \lambda_i' x_a + \epsilon_{ia},$$

where λ_i are the investor embeddings.

- We can also estimate the model in terms of rebalancing.

Methods to extract embeddings

- We consider the following embedding models:

  1. (Supervised) PCA (recommender systems).
  2. Word2Vec.
  3. Models with attention: Transformer models.
     - We build on the BERT architecture and specialize it to holdings data.

(UN)SUPERVISED PCA / RECOMMENDER SYSTEMS

- Recommender systems, with $\theta = (x_a, \lambda_{iq}, \delta_{iq}, \delta_a, \delta_t, \beta_t)$,

$$\min_{\theta} \frac{1-\kappa}{c_h} \sum_{i,a,q} (h_{iaq} - \delta_{iq} - \delta_a - \lambda_{iq}' x_a)^2 + \frac{\kappa}{c_y} \sum_{t,a} (y_{at} - \delta_t - \beta_t' x_a)^2,$$

  where
  - h_{iaq}: Log holdings in quarter q (or active holdings, ...).
  - x_a: Asset embeddings (i.e., recovered characteristics).
  - λ_{iq}: Investor embeddings (i.e., investor tilts).
  - y_{at}: Outcome of interest.

- Analogous to LSA in the NLP literature.[1]

[1] Dumais, Furnas, Landauer, and Deerwester (1988).

Word2Vec

- General approach to estimate language models, such as Word2Vec,[2]
  - Task: Guess masked words.
    - E.g. "Please pass me the _____ and pepper".
  - Use a context window to maximize the probability of a missing word given the context info:

$$\mathbb{P}(w_a \mid w_c) = \frac{\exp(x_a' x_c)}{\sum_b \exp(x_b' x_c)}.$$

[2] Mikolov, Sutskever, Chen, Corrado, Dean (2013a, b).

Word2Vec

- General approach to estimate language models, such as Word2Vec,[2]
  - Task: Guess masked words.
    - E.g. "Please pass me the ____ and pepper".
  - Use a context window to maximize the probability of a missing word given the context info:

$$\mathbb{P}(w_a \mid w_c) = \frac{\exp(x_a' x_c)}{\sum_b \exp(x_b' x_c)}.$$

- Using holdings data:
  - Sentences $\Rightarrow$ Investors.
  - Words $\Rightarrow$ Assets.
  - Task: Guess masked assets.

[2] Mikolov, Sutskever, Chen, Corrado, Dean (2013a, b).

Masked asset modeling

- Example: The ARKK ETF in July 2023:

Holdings Data - ARKK
As of 07/07/2023

ARK INVEST

ARKK
ARK Innovation ETF

	Company	Ticker	CUSIP	Shares	Market Value ($)	Weight (%)
1	TESLA INC	**TSLA**	88160R101	3,496,872	$967,024,982.88	12.43%
2	COINBASE GLOBAL INC -CLASS A	**COIN**	19260Q107	7,945,138	$620,515,277.80	7.98%
3	ROKU INC	**ROKU**	77543R102	8,865,426	$546,110,241.60	7.02%
4	~~ZOOM VIDEO COMMUNICATIONS-A~~	~~**ZM**~~	~~98980L101~~	~~8,258,591~~	~~$534,248,251.79~~	~~6.87%~~
5	UIPATH INC - CLASS A	**PATH**	90364P105	28,152,366	$463,106,420.70	5.95%
6	BLOCK INC	**SQ**	852234103	7,069,493	$456,759,942.73	5.87%
7	EXACT SCIENCES CORP	**EXAS**	30063P105	4,031,264	$368,739,718.08	4.74%
8	UNITY SOFTWARE INC	**U**	91332U101	8,350,868	$338,627,697.40	4.35%
9	SHOPIFY INC - CLASS A	**SHOP**	82509L107	5,430,238	$335,751,615.54	4.32%
10	DRAFTKINGS INC-CL A	**DKNG UW**	26142V105	12,035,607	$303,658,364.61	3.90%

Context and self-attention: A simple example

- So far, we have one x_a per asset, say, Apple, with no context.
- How does attention[4] work?

[4] Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, Polosukhin (2017).

Context and self-attention: A simple example

- So far, we have one x_a per asset, say, Apple, with no context.
- How does attention[4] work?

1. $\mathcal{H}_i$: Stocks in the portfolio of manager i.
2. For stock $a \in \mathcal{H}_i$, compute a similarity score with the other stocks $b \in \mathcal{H}_i$

$$\sigma_{ab} = x_a' x_b.$$

   x_a: Query.
   x_b: Key.

3. Compute the contextualized embedding, x_a^i,

$$x_a^i = \sum_{b \in \mathcal{N}_i} \frac{e^{\sigma_{ab}}}{\sum_{c \in \mathcal{N}_i} e^{\sigma_{ac}}} x_b.$$

   x_b: Value.

[4] Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, Polosukhin (2017).

Self-attention: Example

- Suppose
$$x_a = \begin{bmatrix} x_{a1} \\ x_{a2} \\ x_{a3} \end{bmatrix},$$
where x_{aj} are sub-vectors capturing a firm's industry, reliance on external finance, and supply-chain risk.

- In each quarter, different parts of the embedding vector may be relevant depending on which stocks are held/traded together.

- Similarly, depending on the problem you are studying, you can construct controls depending on what features of firms are relevant in the context of your sample.

Generalizing attention: Transformers

- Transformer models generalize this idea.
  - Query: $q_a = W^Q x_a$.
  - Key: $k_a = W^K x_a$.
  - Value: $v_a = W^V x_a$.
- The contextualized embedding is then computed as

$$x_a^i = \sum_{b \in \mathcal{N}_i} \frac{e^{\sigma_{ab}}}{\sum_{c \in \mathcal{N}_i} e^{\sigma_{ac}}} v_b, \qquad \sigma_{ab} = q_a' k_b.$$

- The matrices W_Q, W_K, and W_V are learned from (lots of) data and determine which aspects of the context are important.

Generalizing attention: Transformers

- Transformer models generalize this idea.
  - Query: $q_a = W^Q x_a$.
  - Key: $k_a = W^K x_a$.
  - Value: $v_a = W^V x_a$.
- The contextualized embedding is then computed as

$$x_a^i = \sum_{b \in \mathcal{N}_i} \frac{e^{\sigma_{ab}}}{\sum_{c \in \mathcal{N}_i} e^{\sigma_{ac}}} v_b, \qquad \sigma_{ab} = q_a' k_b.$$

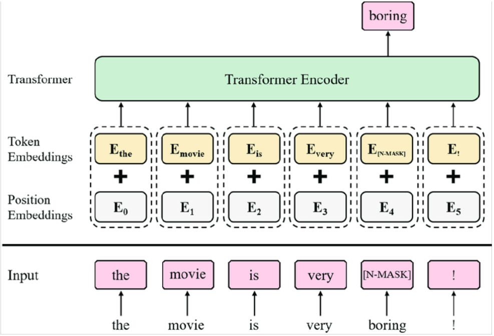
- The matrices W_Q, W_K, and W_V are learned from (lots of) data and determine which aspects of the context are important.
- Features of the full model
  - Stack multiple attention layers with multi-headed attention.
  - Add a feed-forward layer between each self-attention layer:

$$FF(x) = \max(0, xW_1 + b_1)W_2 + b_2,$$

    where the dimensionality of the inner layer $\gg \dim(x)$.
  - Add position embeddings.

BERT: Masked language modeling

- A prime example in NLP is BERT[5] (Bidirectional Encoder Representations from Transformers).
- The model is trained via masked language modeling.
- We estimate a version of a transformer model based on the BERT architecture, AssetBERT.



[5]Devlin, Chang, Lee, Toutanova (2018).

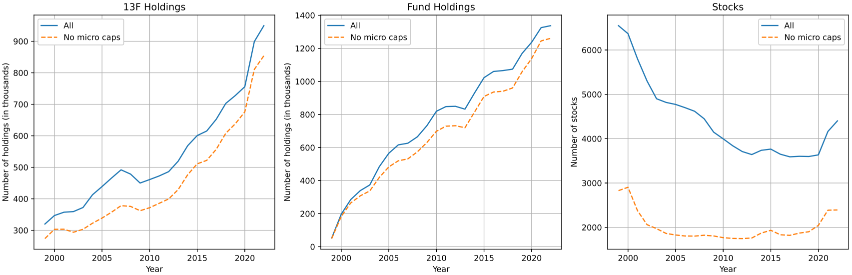
Data

- Holdings data from FactSet:
  - 13F filings.
  - Mutual funds, ETFs, closed-end funds, variable annuity funds.
- Sample construction:
  - 2000.Q1 - 2022.Q4.
  - Remove nano and micro caps.
  - Keep investors (stocks) with at least 20 positions (investors).
- Accounting data and stock returns from CRSP / Compustat, using the Jensen, Kelly, and Pedersen (2023) construction.
- Earnings calls data from FactSet.

Representing firms: The competitors

- Observed characteristics:
  - Market cap, book-to-market, asset growth, profitability, beta, momentum.
- Holdings-based embeddings.
- LLM-based embeddings from Cohere and OpenAI.
  - Cohere:
    - Model: `embed-english-v3.0`.
    - Reduce the dimensionality using UMAP.
  - OpenAI:
    - Model: `text-embedding-3-large`.
    - Download the embeddings for the appropriate size.

Data: 13F and fund holdings



- While the number of stocks has been the declining, the number of investors (and holdings) steadily increased.

Why are benchmarks useful?

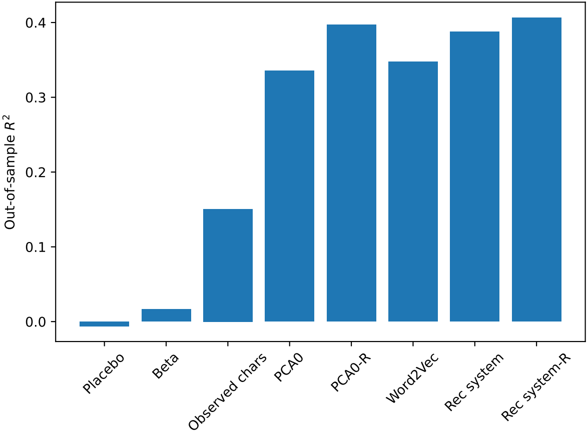
- In ML: Benchmark competitions identify the best performing models, and give metrics for success.
  - E.g. ImageNet to measure improvement in performance in vision tasks.
- We propose to do the same in finance: organize competition every quarter (maybe starting in a few years)
  - Every quarter, researchers would post their predicting software (as a black box).
  - When data are released, we'll see the performance (out of sample) of each model.
- Resembles the current practice, e.g. matching some macro-finance moments, pricing the 25 Fama-French portfolios, ...
- ...Except that the performance here is out of sample (OOS), with new data coming every quarter, so that true OOS performance is easier to evaluate.
- ...and given that the predictions are cross-sectional, just one new quarter is a fairly precise OOS performance test.

Evaluating asset embeddings: Benchmarks

- We consider six benchmarks
  1. Explaining valuations.
  2. Predicting ETF holdings (*ETF*)
  3. Predicting announcement returns.
  4. Missing characteristics.
  5. Predicting demand.
  6. Defining industries (Hoberg and Phillips, 2016) – in progress.

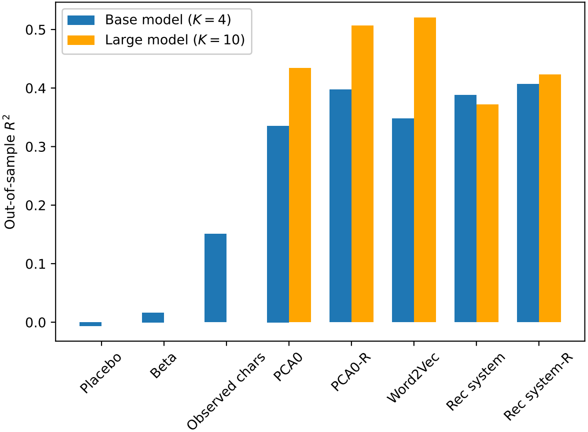
BM 1: Explaining valuations

- Call m_{at} = market equity, b_{at} = book equity.
- Regress $m_{at} = \beta_0 + \beta_1 b_{at} + m_{at}^{\perp}$.
- Fit the valuation residual, $m_{at}^{\perp}$, on x_{at} for 80% of the sample and evaluate, out of sample (OOS), on the remaining 20% using the R^2.



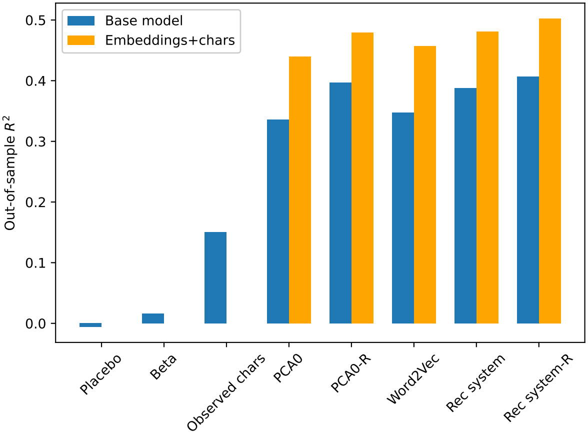
BM 1: Explaining valuations

- Extending the depth of the embeddings tends to improve the fit OOS.



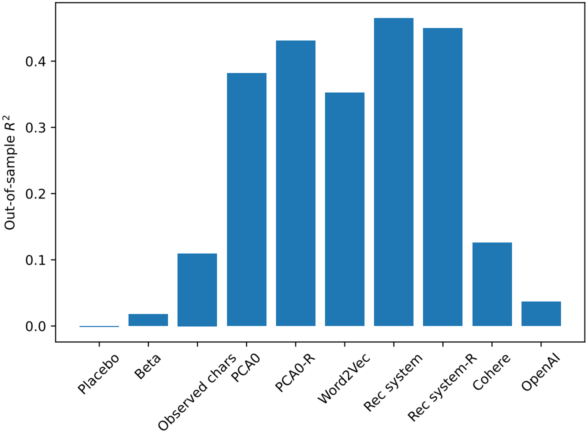
BM 1: Explaining valuations

- Adding characteristics to the base embedding models improves the fit.



BM 1: Explaining valuations

- We compare the observed characteristics and asset embeddings to the text-based embeddings from Cohere and OpenAI.



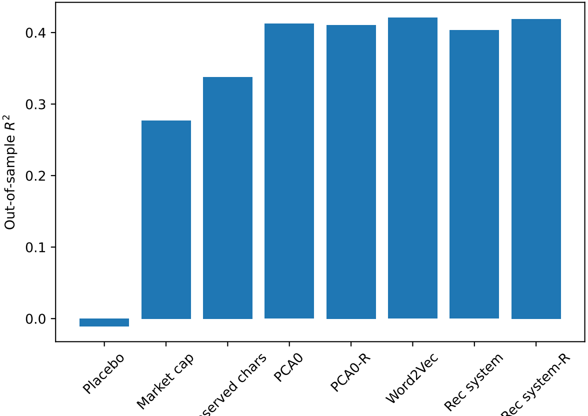
BM 1: Explaining valuations

- Text asset embeddings do understand firms beyond their names, yet names still matter.
- Using the language embeddings from OpenAI, we search for the most similar firms (using cosine similarity).

	OpenAI		
Input company	Apple Inc	Citigroup Inc	Walmart Inc
Rank 1	Appian Corp	Citizens Financial Group Inc	Walgreens Boots
Rank 2	Adobe Inc	Goldman Sachs Group Inc	Home Depot Inc
Rank 3	Interdigital Inc	American International Group Inc	Murphy Usa Inc
Rank 4	Microsoft Corp	Comerica Inc	Amazon Com Inc
Rank 5	Gopro Inc	Cigna Corp New	Qurate Retail Inc
Rank 6	Netapp Inc	Capital One Financial Corp	Big Lots Inc
Rank 7	Intel Corp	Caci International Inc	Burlington Stores
Rank 8	Alphabet Inc	Capital City Bank Group	Dollar Tree Inc
Rank 9	Autodesk Inc	C N O Financial Group Inc	Nordstrom Inc
Rank 10	Appfolio Inc	Jpmorgan Chase & Co	Kohls Corp

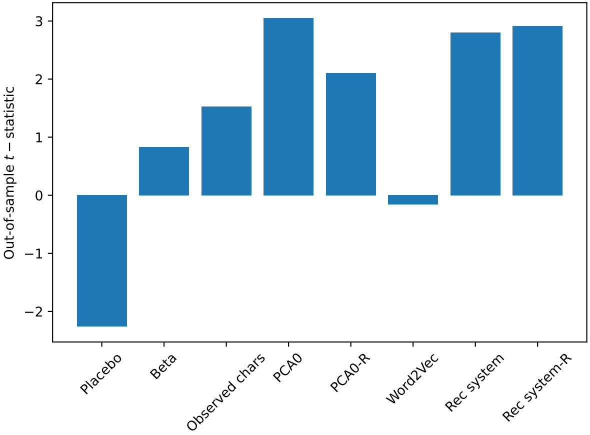
BM 2: ETF SIMILARITY

- We estimate a logit model to predict whether a stock is in a given focused ETF (between 100 and 250 stocks), and compute average performance across ETFs.
- Use 80% of the data (positive and negative samples) to estimate the model and compute the pseudo R2 for the remaining 20% of the data OOS.



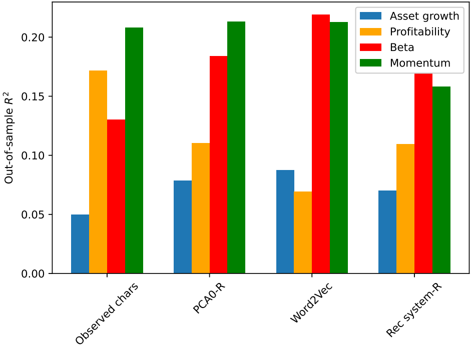
BM 3: Predicting announcement returns

- Regress $CAR3_{at}$ on $x_{a,t-1}$ for the first 80% of announcement days in an earnings quarters and predict the sign of the returns for the remaining 20% OOS. We report the t-stat on slope.



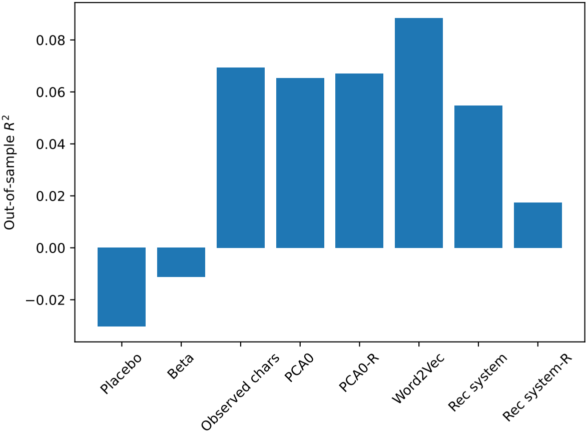
BM 4: Missing characteristics

- Similar to explaining valuations but now with characteristics for asset growth, profitability, momentum, and beta.
  - Use 80% to estimate the link between the characteristic and embeddings to explain 20% OOS.
- To explain missing characteristics, we use other characteristics + size and book/market or large embedding models.
- In progress: Use supervised, regularized recommender systems.



BM 5: Predicting demand

- For investors with more than 250 stocks, we compute their rebalancing (excluding price effects).
- Using 80% of the sample, explain their rebalancing for the remaining 20% OOS.



Interpretability

- How to interpret learned embeddings?
  - For instance, why are some firms close in embedding space (similar $\sigma_{ab} = x_a' x_b$) or changes in embedding space ($\Delta\sigma_{ab}$)?

Interpretability

- How to interpret learned embeddings?
  - For instance, why are some firms close in embedding space (similar $\sigma_{ab} = x_a' x_b$) or changes in embedding space ($\Delta\sigma_{ab}$)?
- We train a RAG-based LLM system for this purpose (RAG: retrieval-augmented generation).
- Main structure:
  1. Create a vector database (Chroma) based on earnings calls.
     - Create chunks of 1,024 tokens with 20 tokens overlap.
     - Embed those using OpenAI's embedding model.
     - Meta data: Firm name, date, industry, and sector codes.
  2. For a given query, embed it, and retrieve vectors from the database using similarity and meta data (LLama Index).
  3. Provide the retrieved chunks as context to answer the query.
- Model details:
  - Embedding model: `text-embedding-3-large`.
  - LLM: `gpt-4-turbo-preview`.

Evaluating transformer models

- AssetBERT generates a distribution over masked assets.
- We consider an initial estimate of the model for a single quarter, 2022.Q4.
  - Context window: 64.
  - Number of layers: 4 (2 attention heads per layer).
- We evaluate the model relative to observed embeddings and the asset embeddings recovered from the recommender system.
- Draw 1,000 managers (with replacement) and, for each manager, mask a stock that we try to predict.

Evaluating transformer models

1. For each investor, fit ranks on embeddings, i.e. estimate $\lambda_{0i}, \lambda_{1i}$ (except masked position):

$$\rho_{ia} = \lambda_{0i} + \lambda_{1i}' x_a + \epsilon_{ia}.$$

2. Predicting a stock at rank ρ, with $\xi_{ia}(\rho) = |\rho - \lambda_{0i} - \lambda_{1i}' x_a|$ and $\gamma_{ia} = \exp(\zeta \xi_{ia}(\rho)) I(a \notin \mathcal{K}_i)$

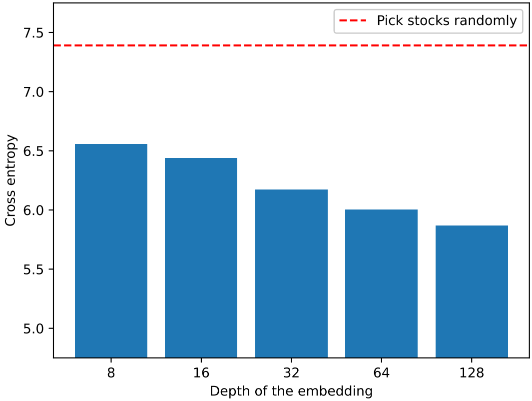
$$\mathbb{P}^{\text{Model}}(\rho_{ia} = \rho \mid \mathcal{J}_{\rho i}) = \frac{\gamma_{ia}}{\sum_b \gamma_{ib}}.$$

3. Cross entropy of the masked words (in set $\mathcal{M}$)

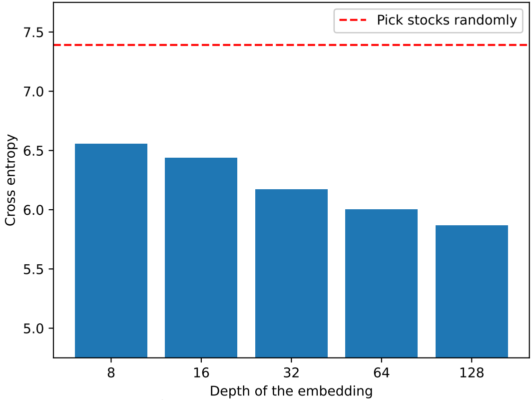
$$CE^{\text{Model}} = -\frac{1}{N} \sum_{a \in \mathcal{M}} \log \mathbb{P}^{\text{Model}}(\rho_{ia} = \rho \mid \mathcal{J}_{\rho i}).$$

4. Model comparison: $CE^{\text{Observed}} - CE^{\text{AssetBERT}}$

Out-of-sample results AssetBERT



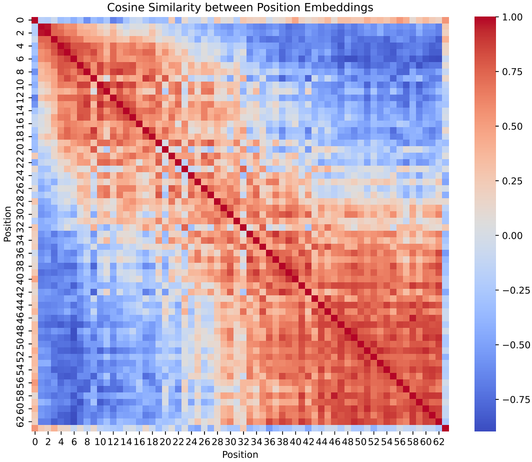
Out-of-sample results AssetBERT



- Relative entropy of
  - Observable characteristics: -0.35 $\Rightarrow$ Likelihood ratio $= 1.41$
  - AssetBERT: -1.67 $\Rightarrow$ Likelihood ratio $= 5.31$
- AssetBERT is 3.71 times more accurate than observable characteristics.

Positional embeddings

- Based on an AssetBERT model with embedding depth of 16, context window of 64 stocks, 4 attention layers, and 2 heads per layer.



Extensions and applications for future work

- Investor embeddings.
  - Characterize investors beyond size, institutional type, ...
- Generative portfolios.
  - Start from salients stocks (e.g., Zoom, Carnival Corp during COVID) and generate a factor.
- Generate stress scenarios.
  - May require other model architecture such diffusion models.
- Other asset classes.
  - Rich holdings data for fixed income markets, derivatives markets, and global equities.

Conclusions

- Recent advances in AI/ML can be applied to economics and finance via asset embeddings.
- We provide a micro foundation for using holdings data.
- We adjust methods that have been successful in related areas (e.g., NLP, vision, ...) to economics:
  - LSA, Word2Vec, Supervised PCA, and Transformer models.