Managing ML Pipelines: Feature Stores and the Coming Wave of Embedding Ecosystems

Feature Store Summit 2021

Outline - Part 1

Feature Stores

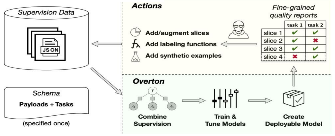
HOPSWORKS KASKADA FEAST

DATA → *feature engineering* → **FEATURE STORE** → *feature serving* → DOWNSTREAM MODELS

self-supervision + large scale training data

Embedding Ecosystems

Overton

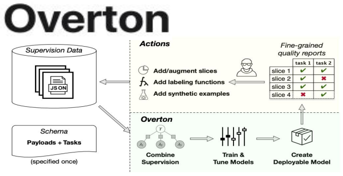


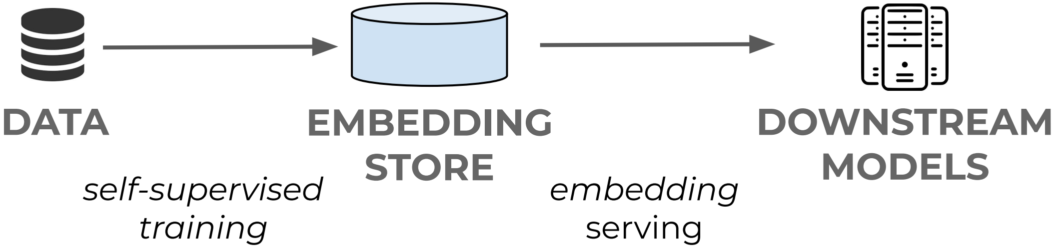
DATA → *self-supervised training* → **EMBEDDING STORE** → *embedding serving* → DOWNSTREAM MODELS

Both systems speak to the importance of reducing engineer effort

Outline - Part 2

Embedding Ecosystems





New Challenges:

Self-Supervised Data Management

Embedding Maintenance

Fine-Grained Evaluation

Feature Store to Embedding Ecosystems

Engineer Workflow *pre Feature Stores (< 2017-8)*

kafka

cassandra

HDFS

STORE and MANAGE DATA

"Pipeline Jungle*"

DEPLOY and MONITOR

*Sculley, David, et al. "Hidden technical debt in machine learning systems." *Neurips* (2015)

The “Pipeline Jungle” Experience

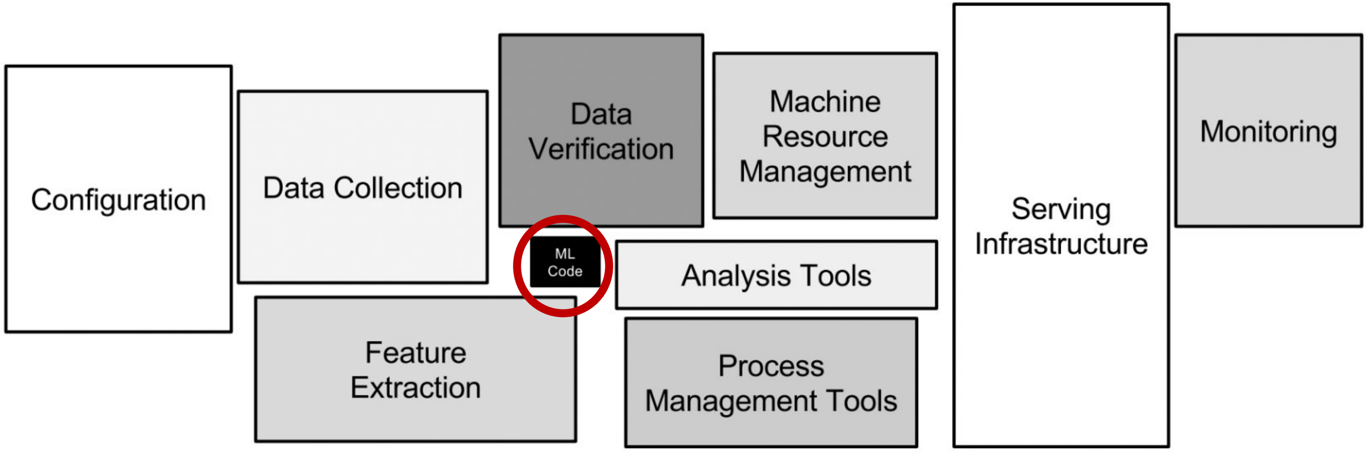


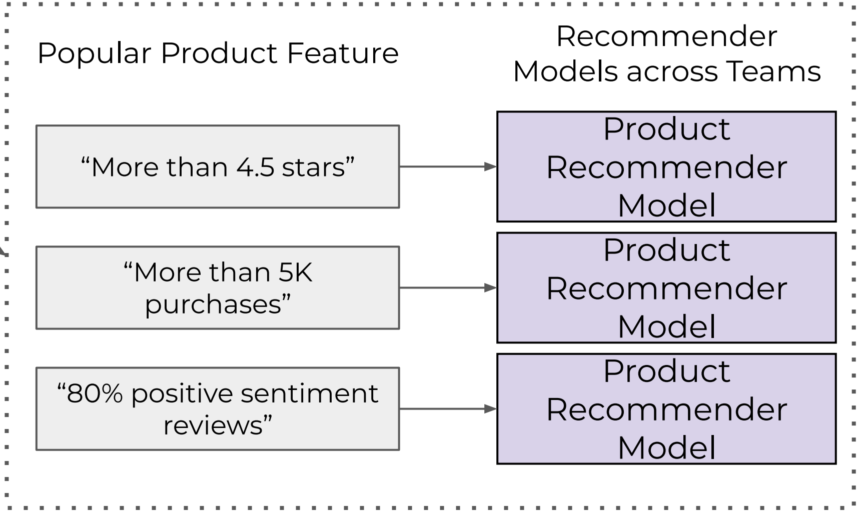
Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

*Sculley, David, et al. "Hidden technical debt in machine learning systems." *Neurips* (2015)

The “Pipeline Jungle” Experience

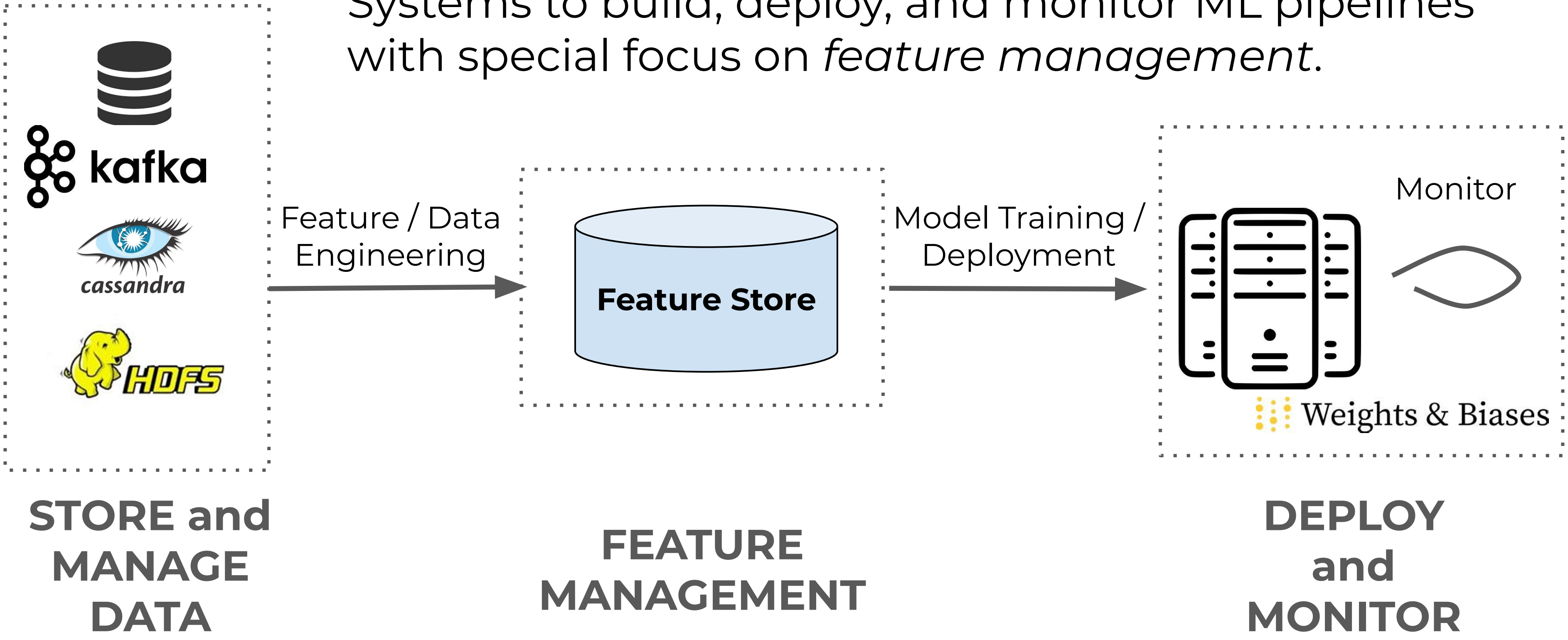
The challenges to deploying a model:

- One-off feature definitions
- Lack of reproducibility
- Inconsistent storage
- No standard evaluations and testing
- Difficult to detect and recover from errors
- ...



Feature Store Solution

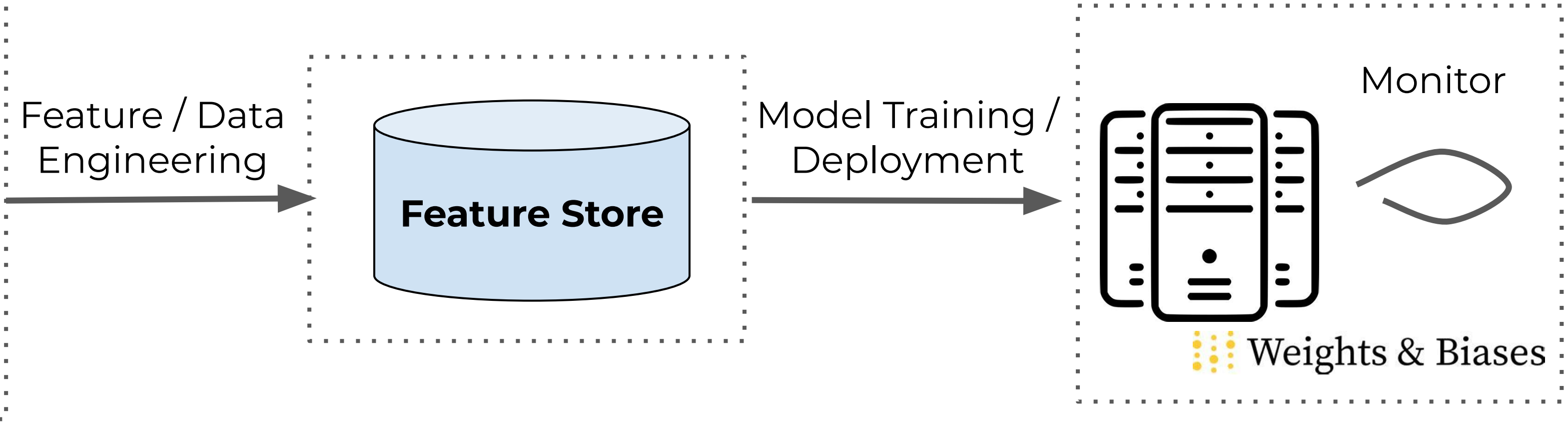
Systems to build, deploy, and monitor ML pipelines with special focus on *feature management*.



Reduction in engineer effort in managing/sharing features

Feature Store Solution

Systems to build, deploy, and monitor ML pipelines with special focus on *feature management*.



STORE and MANAGE DATA

FEATURE MANAGEMENT

DEPLOY and MONITOR

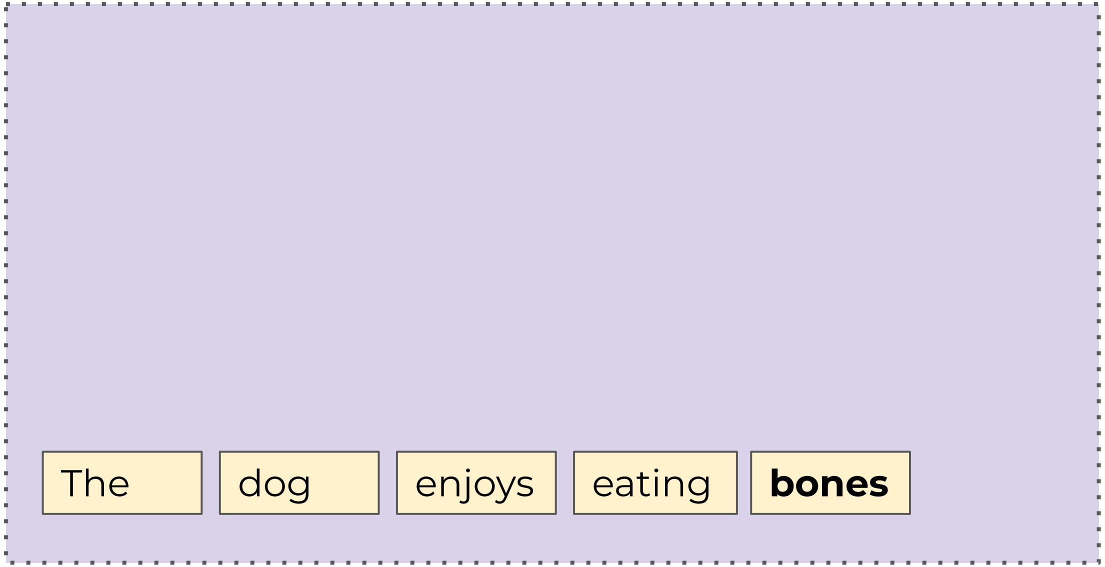
Still needed hand-craft features and labeled training data - expensive!

Enter Self-Supervision

Paradigm where models learn embedding representations of the underlying training data *without* manual labels.

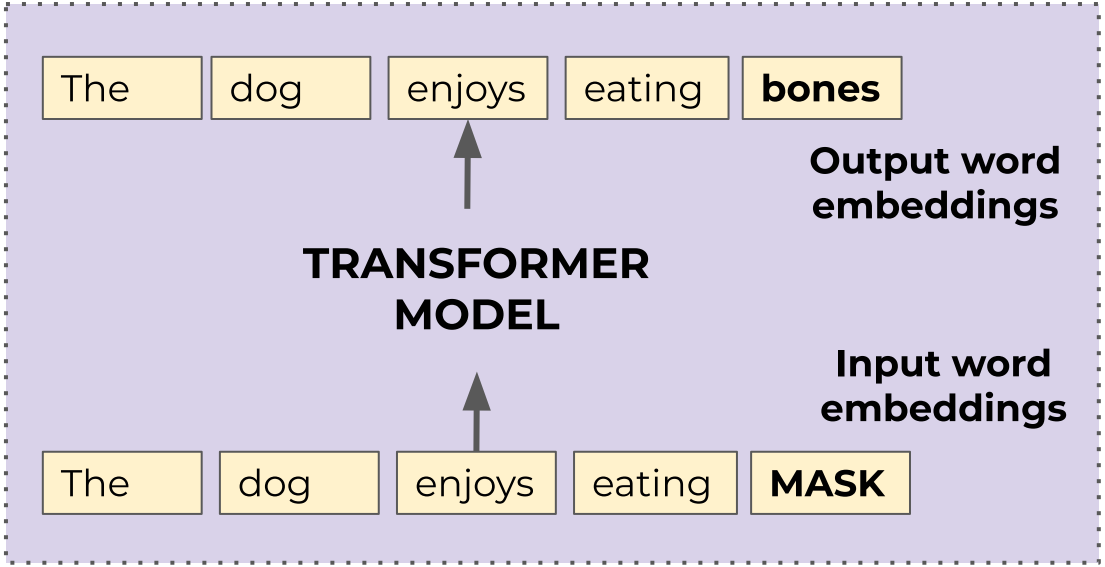
Self-Supervision Example: Transformers and MLM

Learn word embeddings by train a language model to predict a masked word in a given context.



Self-Supervision Example: Transformers and MLM

Learn word embeddings by train a language model to predict a masked word in a given context.

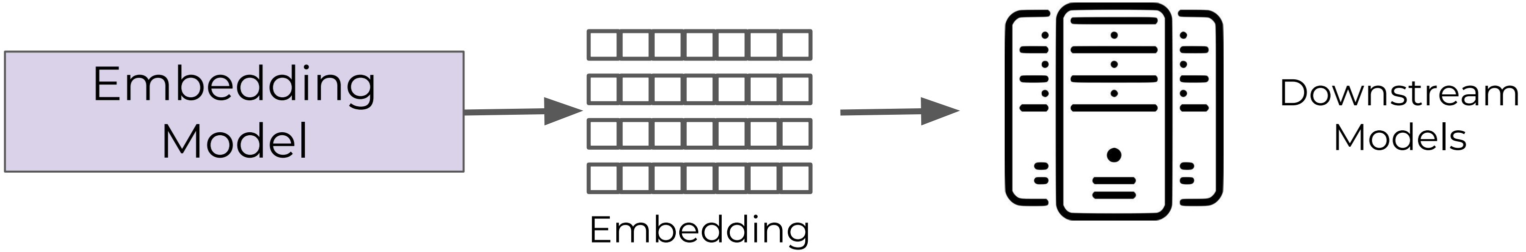


Word embeddings encode contextual information.

Enter Self-Supervision

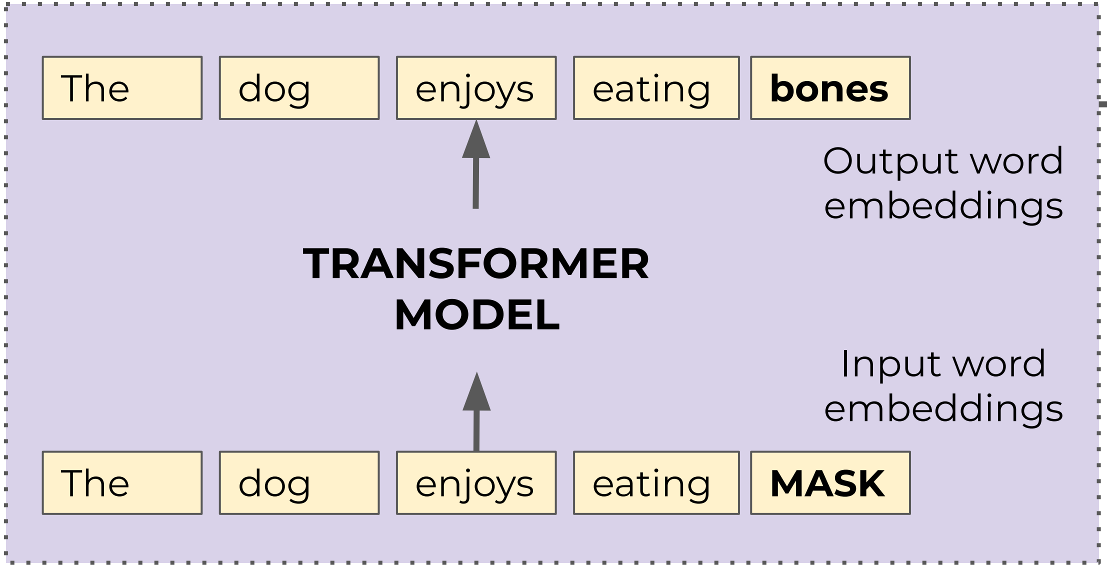
Paradigm where models learn embedding representations of the underlying training data *without* manual labels.

Embeddings are then used in downstream models.



Self-Supervision Example: Transformers and MLM

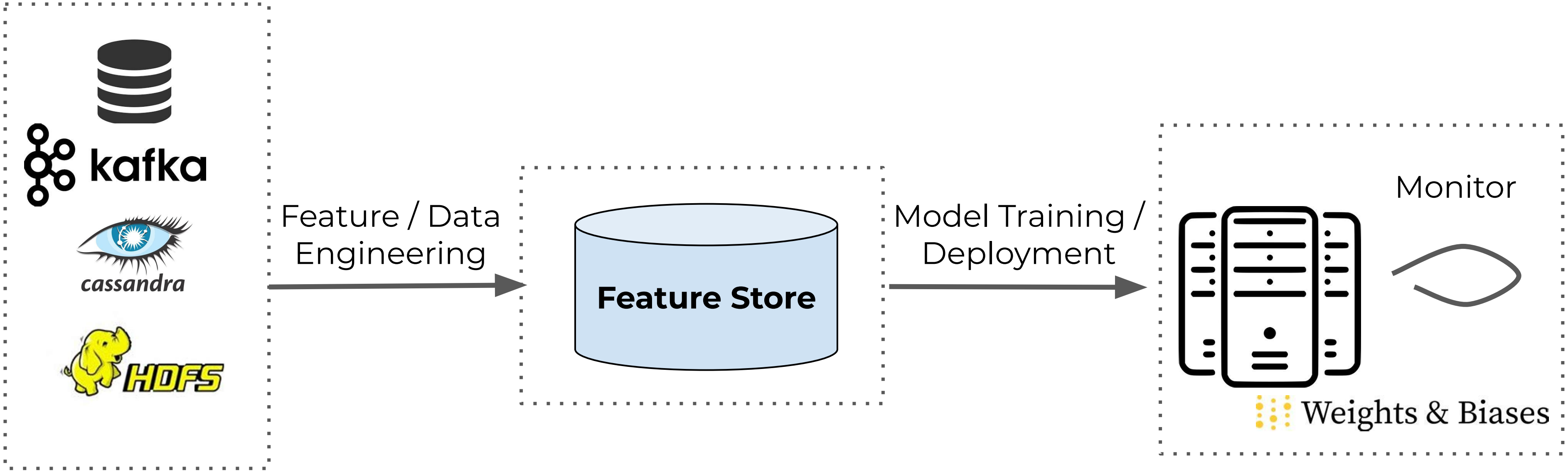
Learn word embeddings by train a language model to predict a masked word in a given context.



Used in downstream sentiment analysis

Word embeddings encode contextual information.

Recall Feature Store Solution



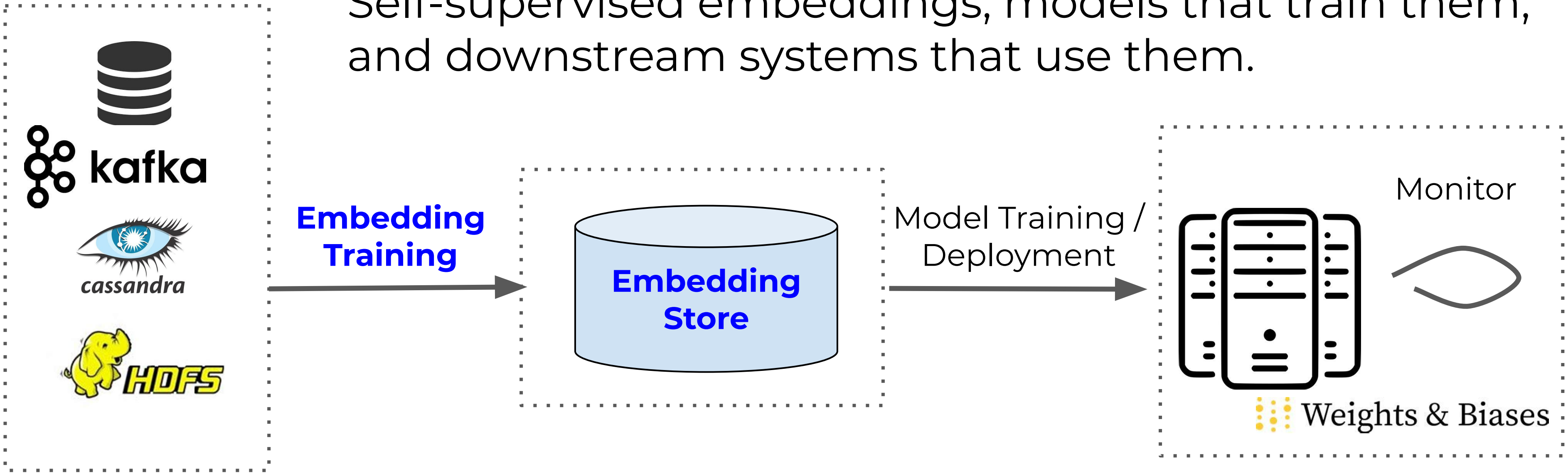
STORE and MANAGE DATA

FEATURE MANAGEMENT

DEPLOY and MONITOR

Embedding Ecosystems

Self-supervised embeddings, models that train them, and downstream systems that use them.



STORE and MANAGE DATA

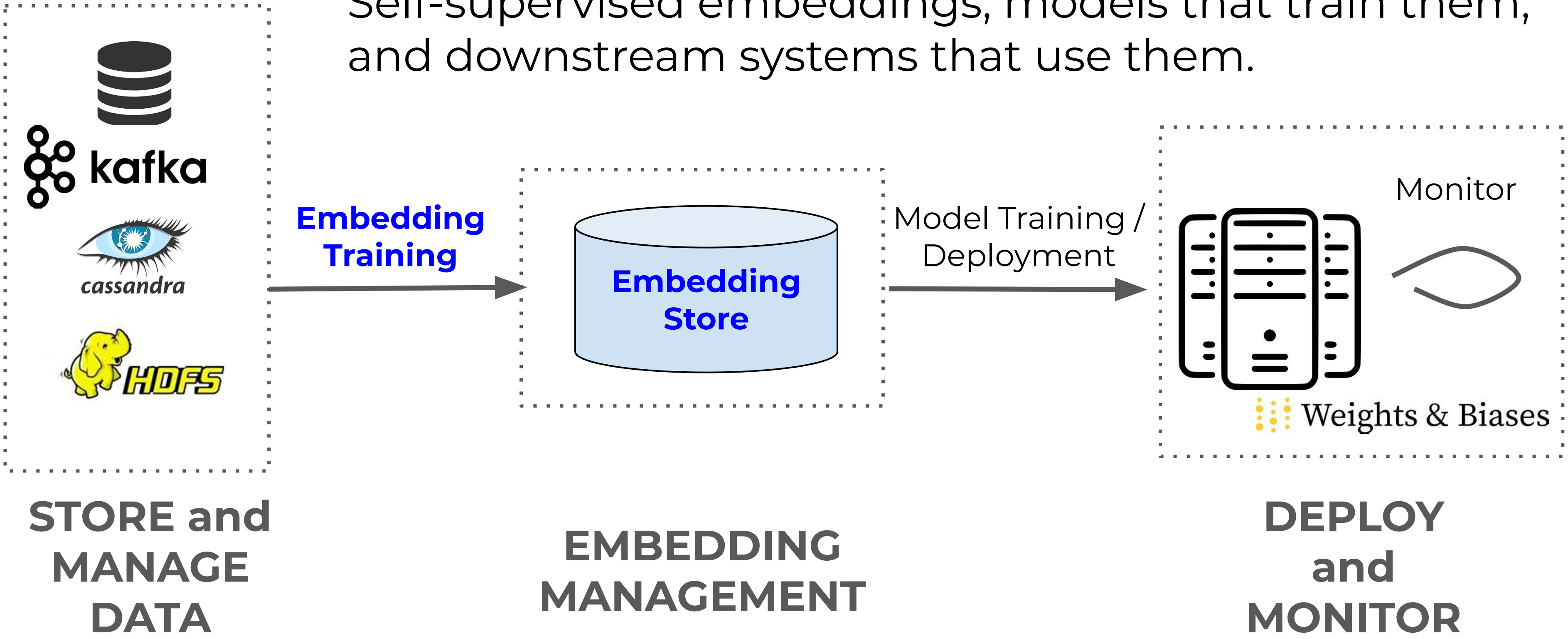
EMBEDDING MANAGEMENT

DEPLOY and MONITOR

No in engineer effort in creating features; easier to maintain models

Embedding Ecosystems

Self-supervised embeddings, models that train them, and downstream systems that use them.



Open challenges in data, embedding, and model management

Embedding Ecosystems

Challenges of Embedding Ecosystems

Self-Supervised Data Management

Embedding Maintenance

Fine-Grained Evaluation

Challenges of Embedding Ecosystems

Self-Supervised Data Management

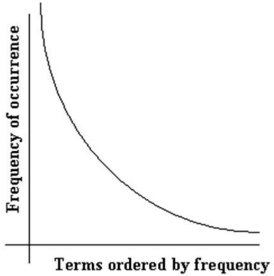
- Rare item biases
- Multi-modal
- Large-scale

Embedding Maintenance

- Embedding updates
- Provenance
- Search

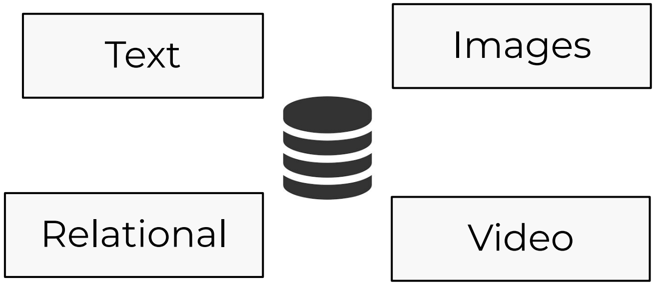
Fine-Grained Evaluation

- User-in-the-loop
- Slice finding
- Model patching



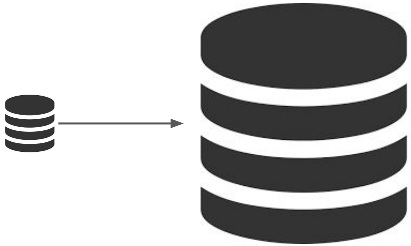
Data Bias

How to overcome systematic biases in unlabeled data?



Multi-Modal

How to support integrating heterogeneous sources?



Large-Scale

How to support data exploration and training over PB?

Challenges of Embedding Ecosystems

Self-Supervised Data Management

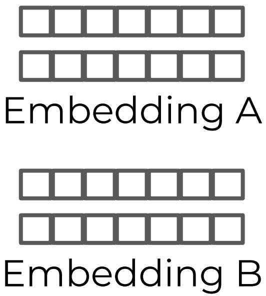
- Rare item biases
- Multi-modal
- Large-scale

Embedding Maintenance

- Provenance
- Search
- Embedding updates

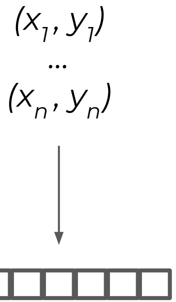
Fine-Grained Evaluation

- User-in-the-loop
- Slice finding
- Model patching



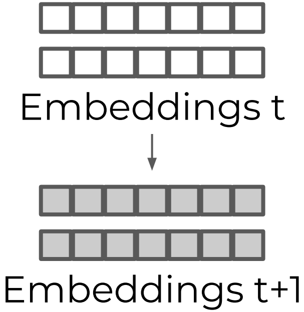
Search

What set of embeddings are best for a specific task?



Provenance

What data had the most “impact” on these embeddings?



Updates

How to update embeddings when changes to data?

Challenges of Embedding Ecosystems

Self-Supervised Data Management

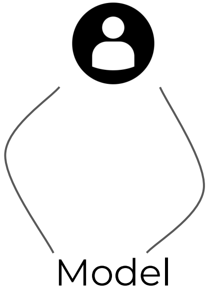
- Rare item biases
- Multi-modal
- Large-scale

Embedding Maintenance

- Provenance
- Search
- Embedding updates

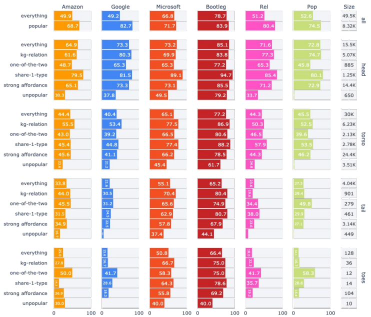
Fine-Grained Evaluation

- User-in-the-loop
- Slice finding
- Model patching



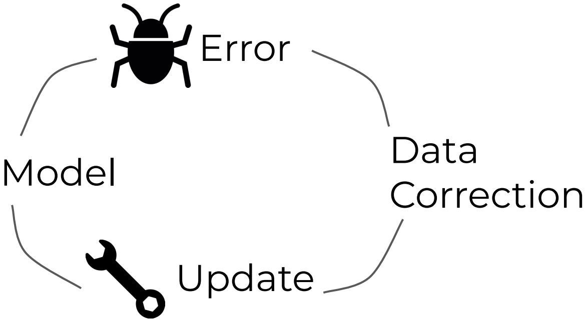
User-in-the-Loop

What are the right data structures to support interactive analysis?



Slice Finding

What are the current failure modes?



Model Patching

How to correct for errors in models?

Challenges of Embedding Ecosystems

Self-Supervised Data Management

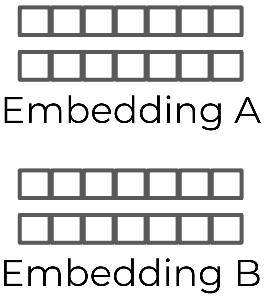
- Rare item biases
- Multi-modal
- Large-scale

Embedding Maintenance

- Provenance
- Search
- Embedding updates

Fine-Grained Evaluation

- User-in-the-loop
- Slice finding
- Model patching



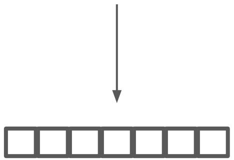
Search

What set of embeddings are best for a specific task?

(x_1, y_1)

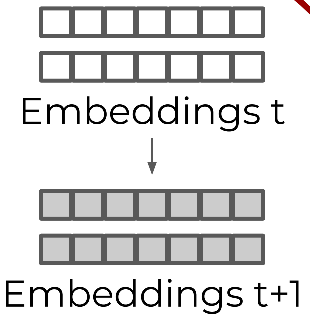
…

(x_n, y_n)



Provenance

What data had the most “impact” on these embeddings?



Updates

How to update embeddings when changes to data?

Deep Dive: Embedding Updates

Grounding Use Case: Named Entity Disambiguation

Map “strings to things” in a knowledge base.

How tall is *Lincoln*?

Q91

LINCOLN

Embeddings key part of assistant, search, and information extraction

Entities Are Continuously Changing

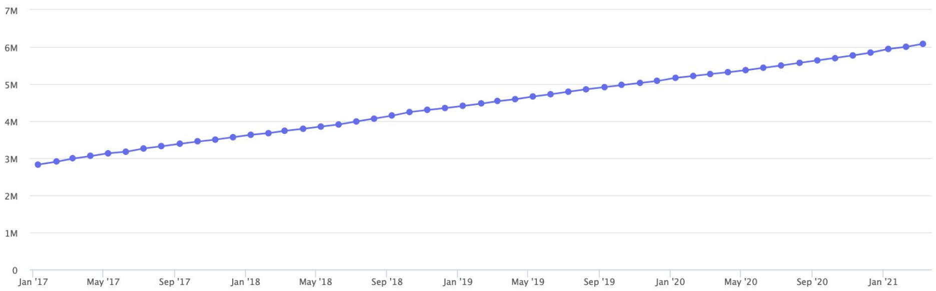
Annual English Wikipedia Growth Rate

Date	Article count	Increase during preceding year	% Increase during preceding year	Doubling time (in years and days rounded up)	Average increase per day during preceding year
2002-01-01	19,700	19,700	—	—	54
2003-01-01	96,500	76,800	390%	160 days	210
2004-01-01	188,800	92,300	96%	377 days	253
2005-01-01	438,500	249,700	132%	301 days	682
2006-01-01	895,000	456,500	104%	355 days	1251
2007-01-01	1,560,000	665,000	74%	342 days	1822
2008-01-01	2,153,000	593,000	38%	1 year, 302 days	1625
2009-01-01	2,679,000	526,000	24%	2 years, 326 days	1437
2010-01-01	3,144,000	465,000	17%	4 years, 29 days	1274
2011-01-01	3,518,000	374,000	12%	5 years, 284 days	1025
2012-01-01	3,835,000	317,000	9%	7 years, 257 days	868
2013-01-01	4,133,000	298,000	8%	8 years, 243 days	814
2014-01-01	4,413,000	280,000	7%	9 years, 330 days	767
2015-01-01	4,682,000	269,000	6%	11 years, 202 days	736
2016-01-01	5,045,000	363,000	8%	8 years, 243 days	995
2017-01-01	5,321,200	276,200	7%	9 years, 330 days	755
2018-01-01	5,541,900	220,700	4.5%	15 years, 148 days	605
2019-01-01	5,773,600	231,700	4.2%	16 years, 310 days	635
2020-01-01	5,989,400	215,800	3.75%	20 years, 11 days	591
2021-01-01	6,219,700	230,300	3.8%	20 years	629
2021-10-01	6,386,116	**166,388**[a]	—	—	**608**[a]

[a] Calculated live, so far, as only for partial year.

~630 new pages every day

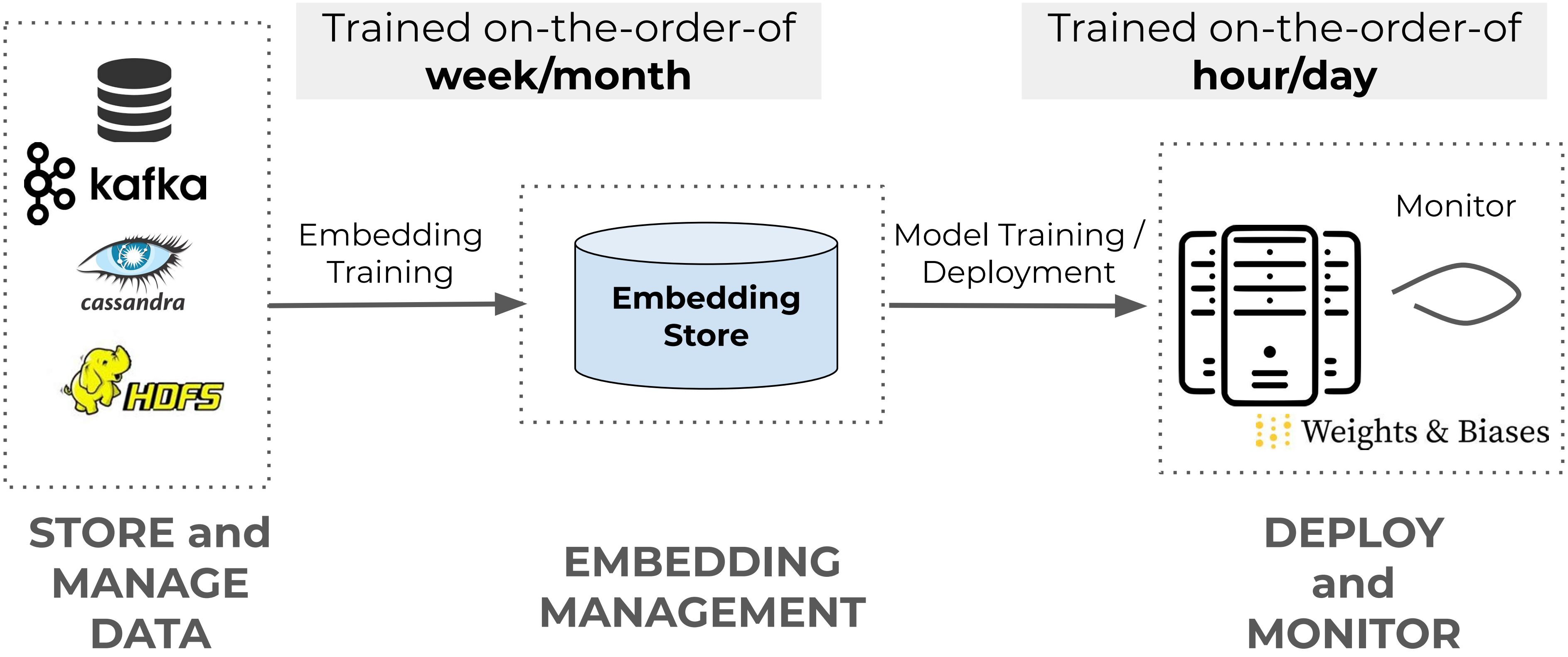
Number Amazon Sellers



~73 new sellers every hour

*https://www.marketplacepulse.com/amazon/number-of-sellers, https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia

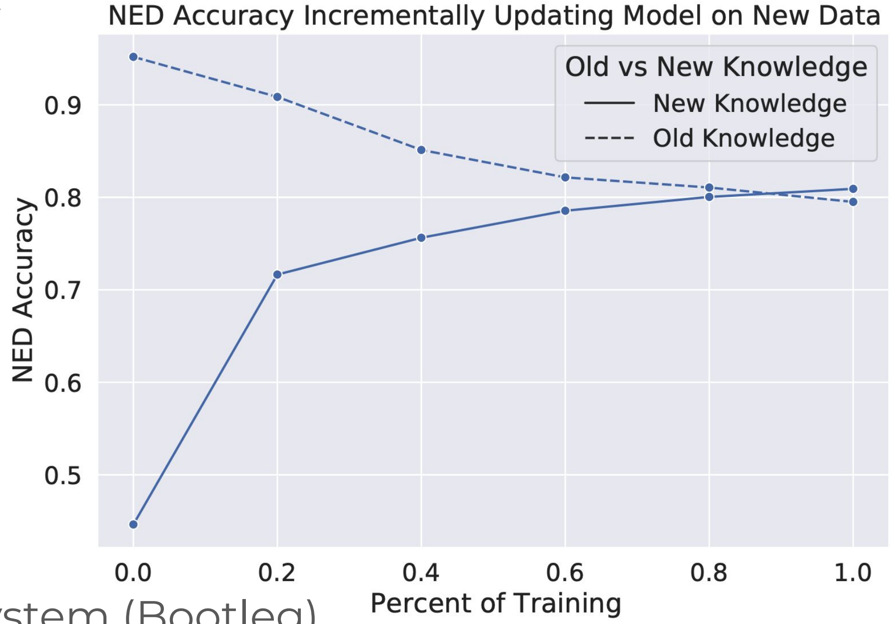
Update Frequency



Need to continuously update embedding models to keep up.

Forgetting Old Knowledge

What happens when we continuously train a model on stream of new knowledge?

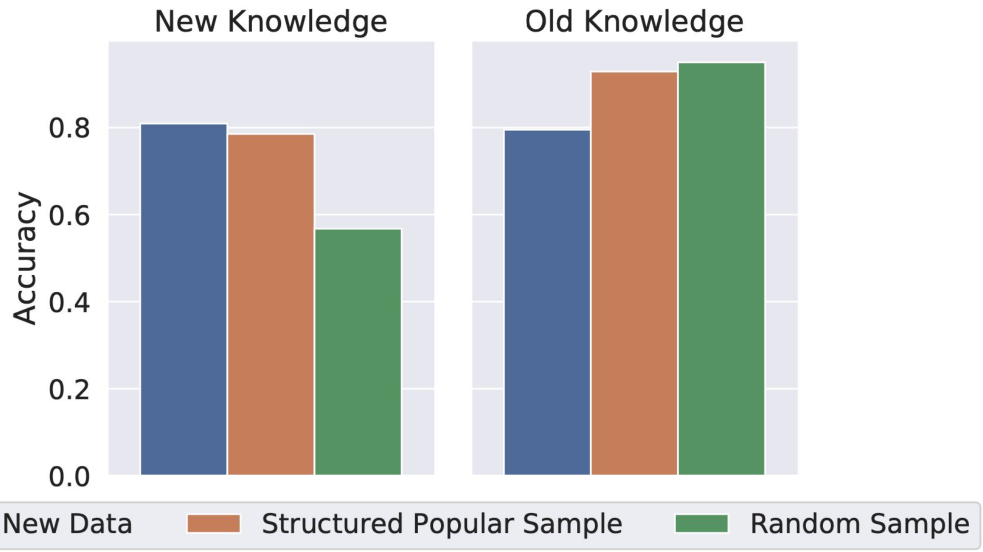


Case-Study: NED System (Bootleg)

Streaming update methods forget old information, especially popular entities.

Tail Insight

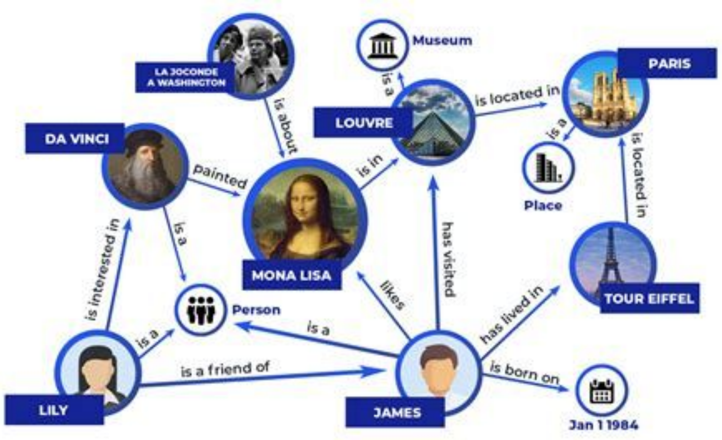
Tail entities leverage structured knowledge
-> Update all popular entities *and structured metadata*



Sample balances new and old knowledge.

Updating Entity Knowledge Take Away

Entities are continuously changed and embedding models need to be quickly updated to maintain quality

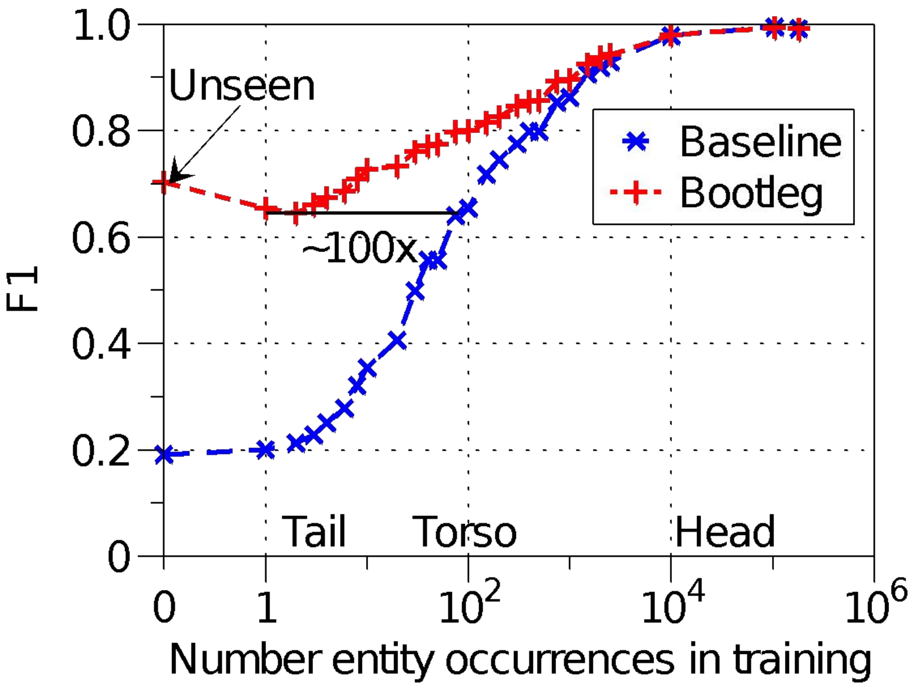


Take advantage of the integrated structured and unstructured data when selecting update data.

Challenge 1: Tail Bias

Tail Challenge

Impossible to scale the data to memorize all patterns needed for rare entities.

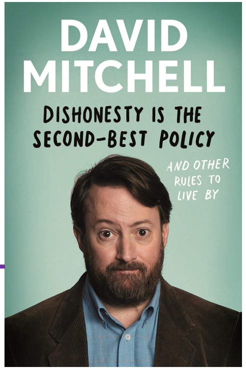


Subtle reasoning clues are needed for the tail!
(+40 F1 points by encoding these reasoning patterns)

Reasoning over Relationships


Victoria Mitchell
(poker player, writer)

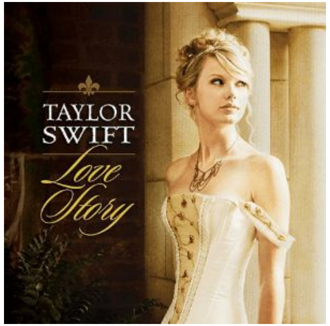
spouses



David Mitchell


Victoria
Mitchell
(runner)

David and Victoria Mitchell added spice to their marriage



Love Story by Taylor Swift

Love Story by Andy Williams

Play Love Story by Williams

* Images from Wikipedia

Reasoning over Types

How tall is Lincoln?

People have heights, not places or brands

What is the cheapest Lincoln?

LINCOLN

Brands have prices, not places or people

How many people are in Lincoln?

Places have populations, not people or brands

Bootleg: Tackles the Tail with Structural Knowledge

BOOTLEG

Key Idea: reasoning over *type* and *relationship* signals can resolve unseen entities.

Implementation: use *embeddings* to teach a model to reason over types and relationships.

Orr, Laurel, et al. "Bootleg: Chasing the tail with self-supervised named entity disambiguation." *arXiv preprint arXiv:2010.10363* (2020).

Bootleg: Tail Performance

On the head, BERT-based baseline performs ~ 5 F1 points of Bootleg.

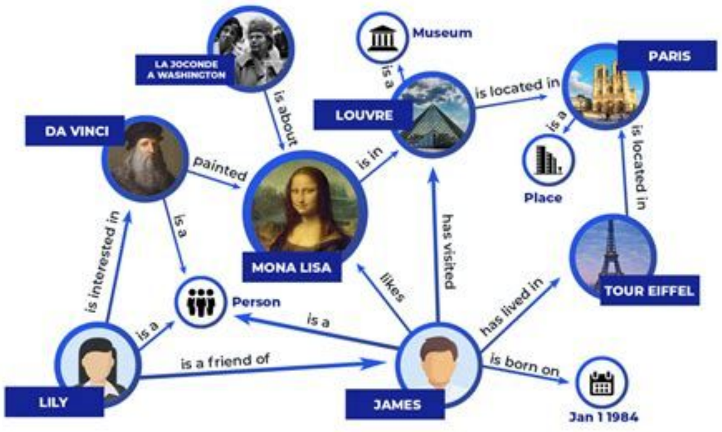
On the tail, Bootleg outperforms baseline by > 40 F1 points!

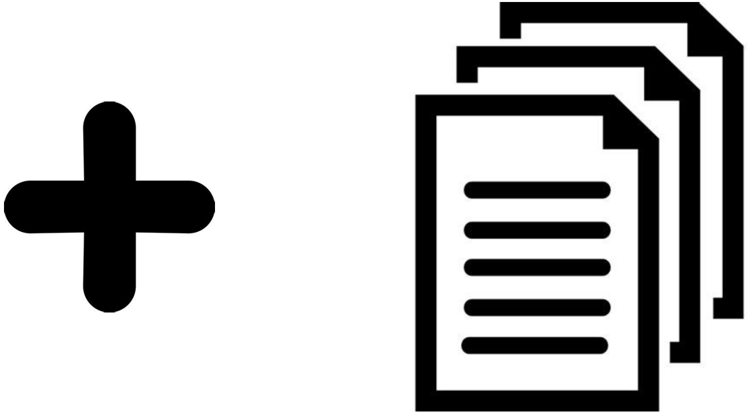
Evaluation Set	BERT NED Baseline	Bootleg	# Examples
All	85.9	**91.3**	4,066K
Torso Entities	79.3	**87.3**	1,912K
Tail Entities	27.8	**69.0**	163K
Unseen Entities	18.5	**68.5**	10K

Performance results on Wikipedia dataset.

Self-Supervised Data Take Away

Self-supervised data does not well represent tail distributions -> embeddings may not be high quality for rare entities





Integrating structured knowledge with unstructured data allows for better generalization to the tail.