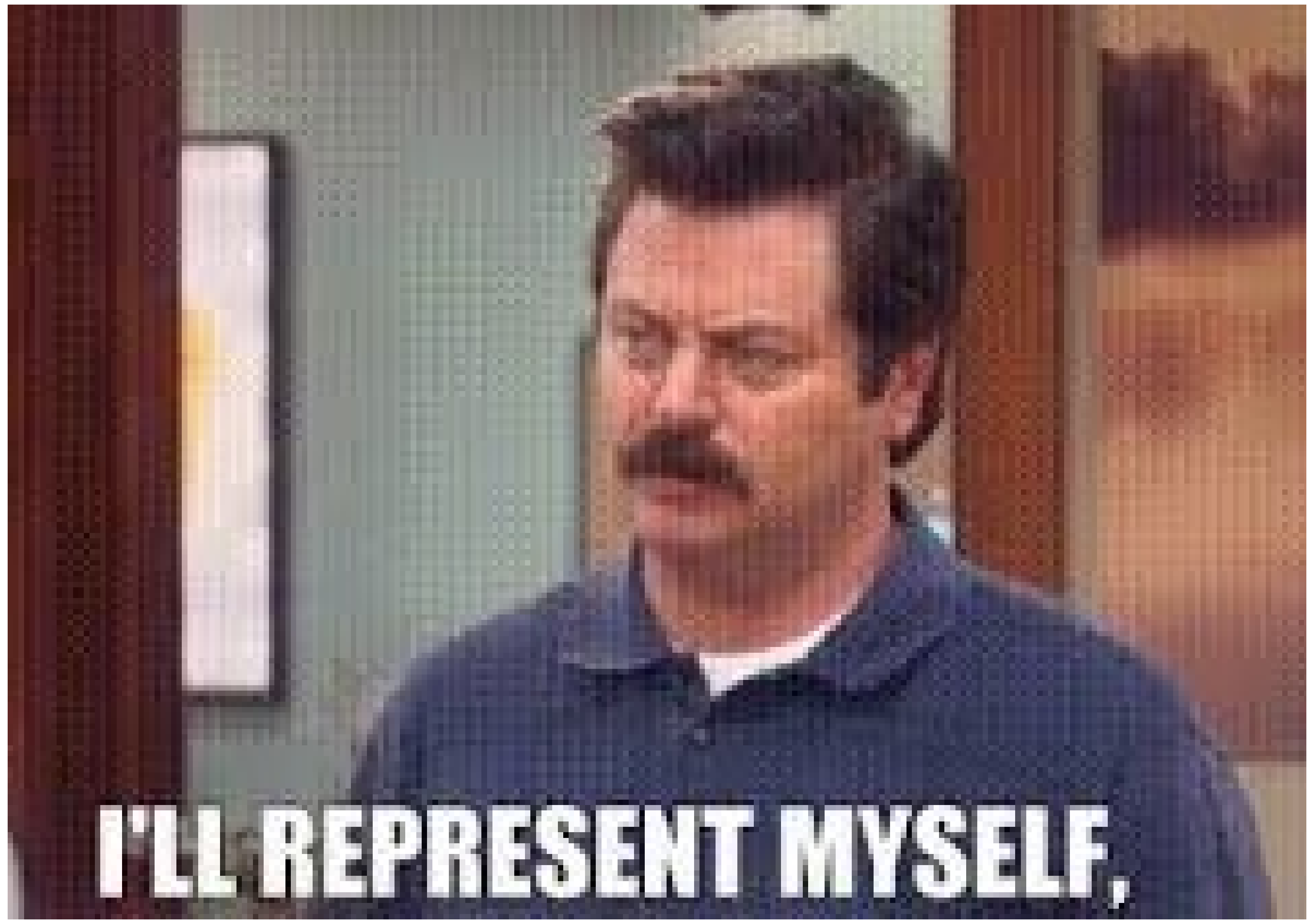
CMSC 475/675 Neural Networks

Lecture 6: Representation Learning & Generative Models



UMBC

Some VAE slides are from Ranjay Krishna

Machine Learning Problems

SUPERVISED LEARNING

- Training time
  - data :

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y^{(t)} \leftarrow p(\mathbf{x}, y)$$

- Test time
  - data :

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y^{(t)} \leftarrow p(\mathbf{x}, y)$$

Example

$Input$: $x^{(t)}$ is an image

$Output$: $y^{(t)}$ is an image category

MULTITASK LEARNING

- Training time
  - data :

$$\{\mathbf{x}^{(t)}, y_1^{(t)}, \ldots, y_M^{(t)}\}$$

  - setting :

$\mathbf{x}^{(t)}, y_1^{(t)}, \ldots, y_M^{(t)}$ ⇠

$$p(\mathbf{x}, y_1, \ldots, y_M)$$

- Test time
  - data :

$$\{\mathbf{x}^{(t)}, y_1^{(t)}, \ldots, y_M^{(t)}\}$$

  - setting :

$\mathbf{x}^{(t)}, y_1^{(t)}, \ldots, y_M^{(t)}$ ⇠

$$p(\mathbf{x}, y_1, \ldots, y_M)$$

Example

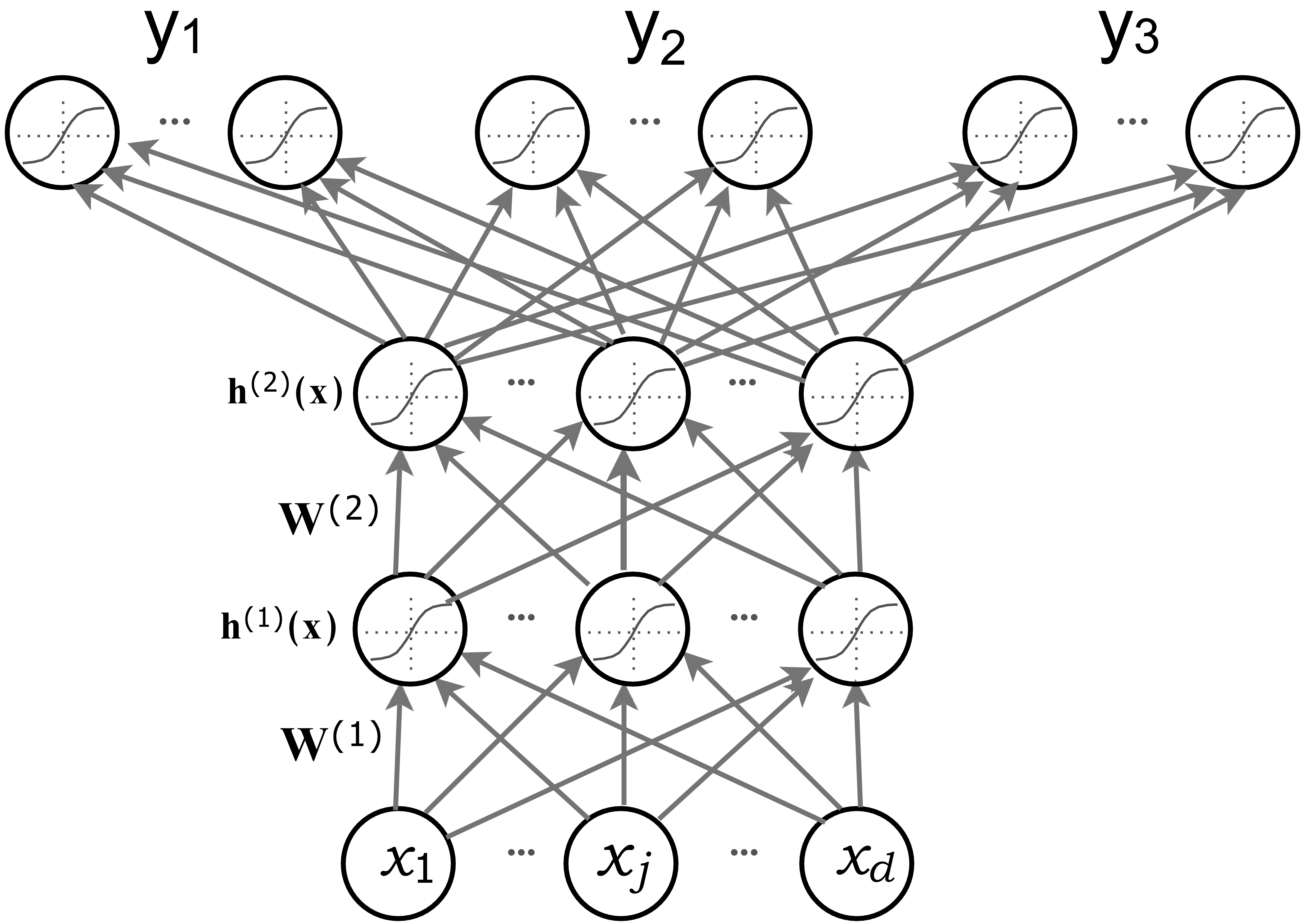
$Input$: $x^{(t)}$ is an image

$Outputs$:

- $y_1^{(t)}$: image category
- $y_1^{(t)}$: object detection
- $y_1^{(t)}$: depth estimation
- $y_1^{(t)}$: semantic segmentation
- …

MULTITASK LEARNING

Topics: multitask learning



DOMAIN ADAPTATION

- Training time
  - data :

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

$$\{\bar{\mathbf{x}}^{(t')}\}$$

  - setting :

$$\mathbf{x}^{(t)} \dashleftarrow p(\mathbf{x})$$

$$y^{(t)} \dashleftarrow p(y|\mathbf{x}^{(t)})$$

$$\bar{\mathbf{x}}^{(t)} \dashleftarrow q(\mathbf{x}) \uparrow \ p(\mathbf{x})$$

- Test time
  - data :

$$\{\bar{\mathbf{x}}^{(t)}, y^{(t)}\}$$

  - setting :

$$\bar{\mathbf{x}}^{(t)} \dashleftarrow q(\mathbf{x})$$

$$y^{(t)} \dashleftarrow p(y|\bar{\mathbf{x}}^{(t)})$$

- Example
  - classify sentiment (positive vs negative)
  - in reviews of different products
  - training on Amazon Reviews but testing on Yelp Reviews

ONE-SHOT LEARNING

- Training time
  - data :

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y^{(t)} \dashleftarrow p(\mathbf{x}, y)$$

subject to $y^{(t)}\ 2\ \{1, \ldots, C\}$

- Test time
  - data :

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y^{(t)} \dashleftarrow p(\mathbf{x}, y)$$

subject to $y^{(t)}\ 2\ \{C + 1, \ldots, C + M\}$

  - side information :
    - a single labeled example from each of the M new classes

- Example
  - recognizing a person based on a single picture of him/her

ZERO-SHOT LEARNING

- Training time
  - data :

    $$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

  - setting :

    $$\mathbf{x}^{(t)}, y^{(t)} \leftarrow\cdots p(\mathbf{x}, y)$$

    subject to $y^{(t)}\ 2\ \{1, \ldots, C\}$
  - side information :
    - description vector z_c of each of the C classes

- Test time
  - data :

    $$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

  - setting :

    $$\mathbf{x}^{(t)}, y^{(t)} \leftarrow\cdots p(\mathbf{x}, y)$$

    subject to $y^{(t)}\ 2\ \{C + 1, \ldots, C + M\}$
  - side information :
    - description vector z_c of each of the new M classes

- Example
  - recognizing an object based on a worded description of it

SEMI-SUPERVISED LEARNING

- Training time
  - data :

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

$$\{\mathbf{x}^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y^{(t)} \dashleftarrow p(\mathbf{x}, y)$$

$$\mathbf{x}^{(t)} \dashleftarrow p(\mathbf{x})$$

- Test time
  - data :

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y^{(t)} \dashleftarrow p(\mathbf{x}, y)$$

UNSUPERVISED LEARNING

- Training time
  - data :

$$\{\mathbf{x}^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)} \leftarrow p(\mathbf{x})$$

- Test time
  - data :

$$\{\mathbf{x}^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)} \leftarrow p(\mathbf{x})$$

How can we train models “unsupervised”?

How can we train models “unsupervised”?

This is the focus of representation learning

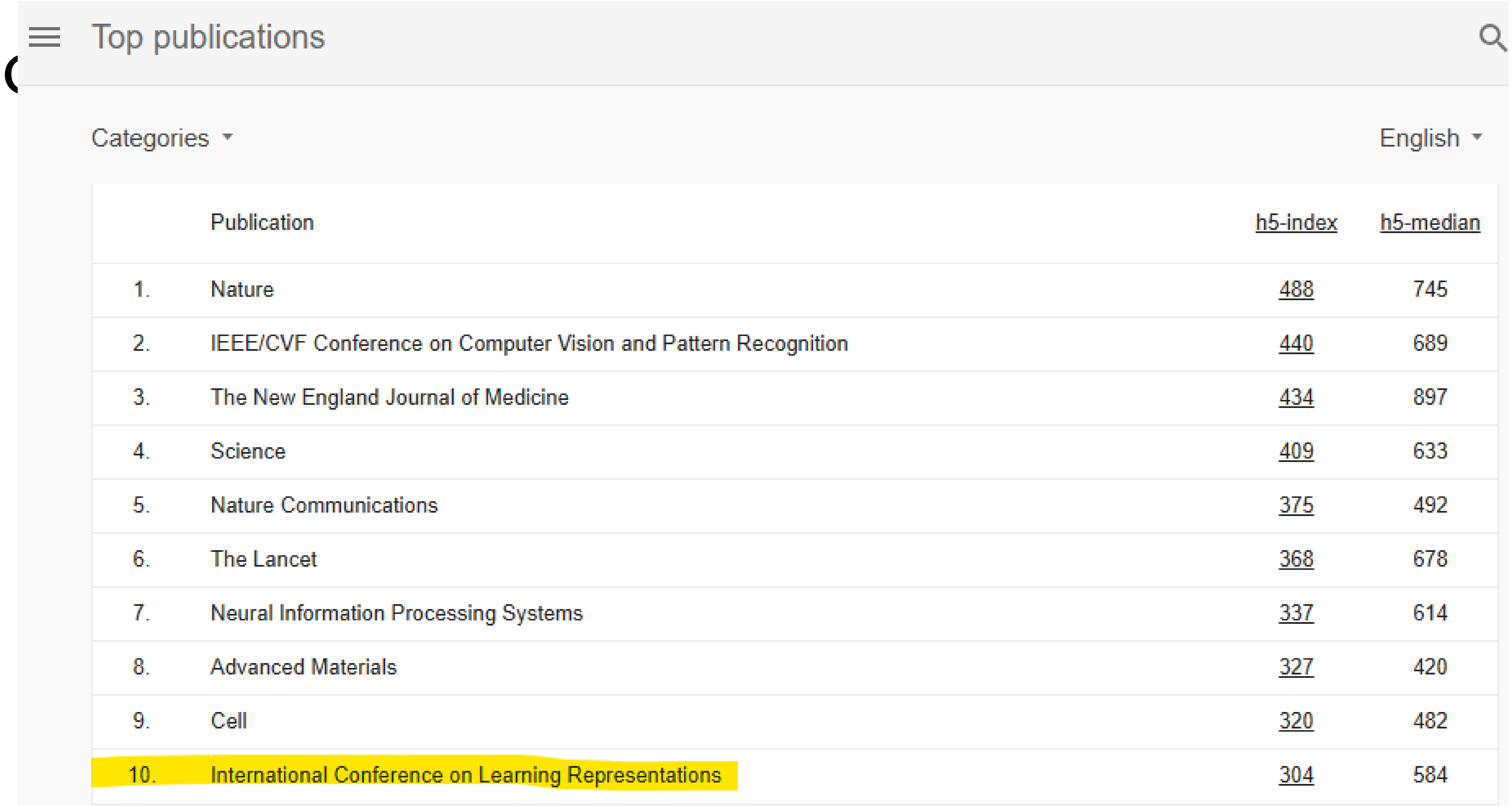
How can we train models "unsupervised"?

This is the focus of representation learning

There's an entire co

ICLR has become one of the top CS and Engineering (not just AI) publication although it just started in 2013.

In fact top-10 in ALL OF SCIENCE



	Publication	h5-index	h5-median
1.	Nature	488	745
2.	IEEE/CVF Conference on Computer Vision and Pattern Recognition	440	689
3.	The New England Journal of Medicine	434	897
4.	Science	409	633
5.	Nature Communications	375	492
6.	The Lancet	368	678
7.	Neural Information Processing Systems	337	614
8.	Advanced Materials	327	420
9.	Cell	320	482
10.	International Conference on Learning Representations	304	584

How

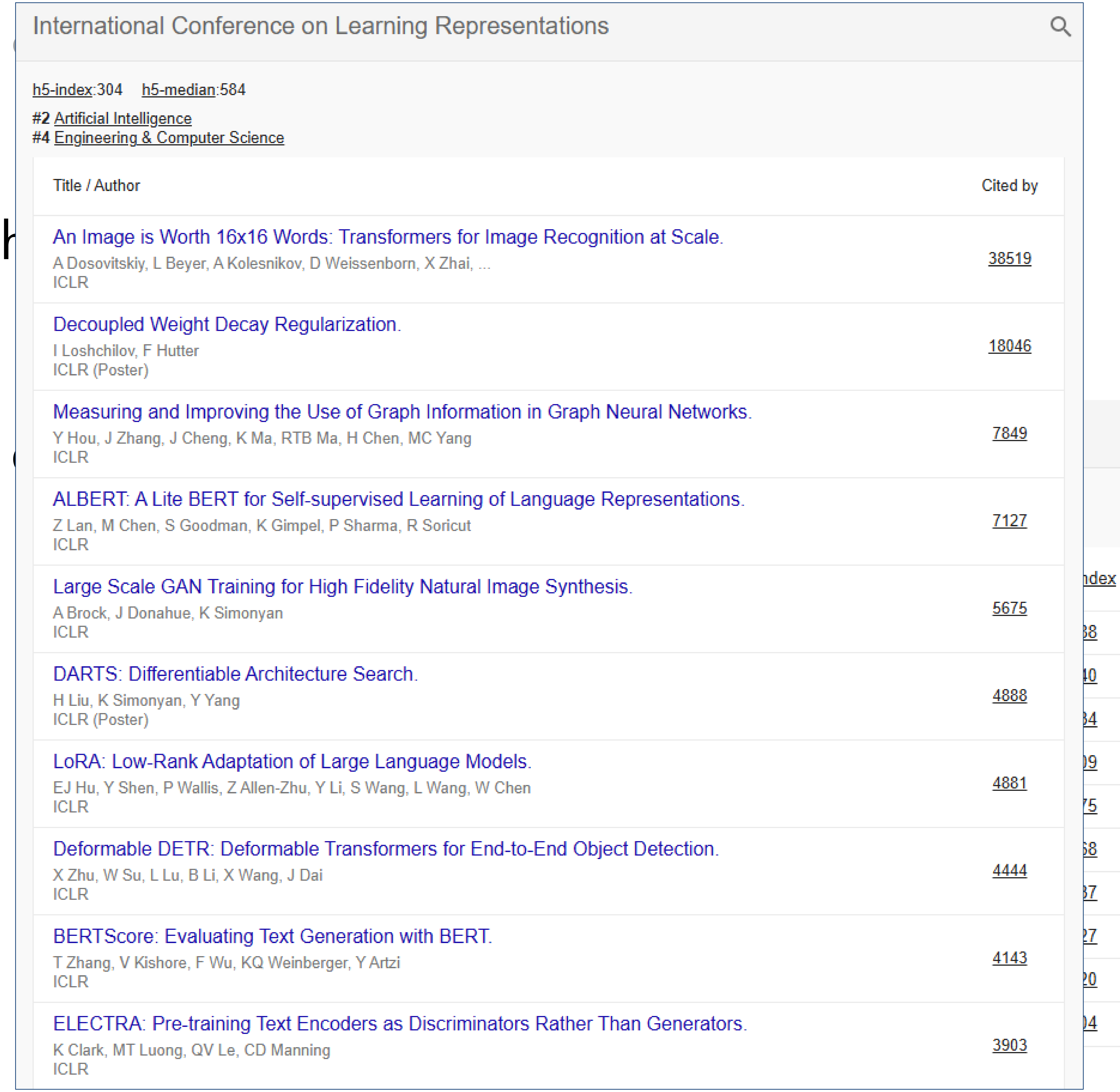
This is th

There's an entire

ICLR

ICLR has become one of the top CS and Engineering (not just AI) publication although it just started in 2013.

In fact top-10 in ALL OF SCIENCE

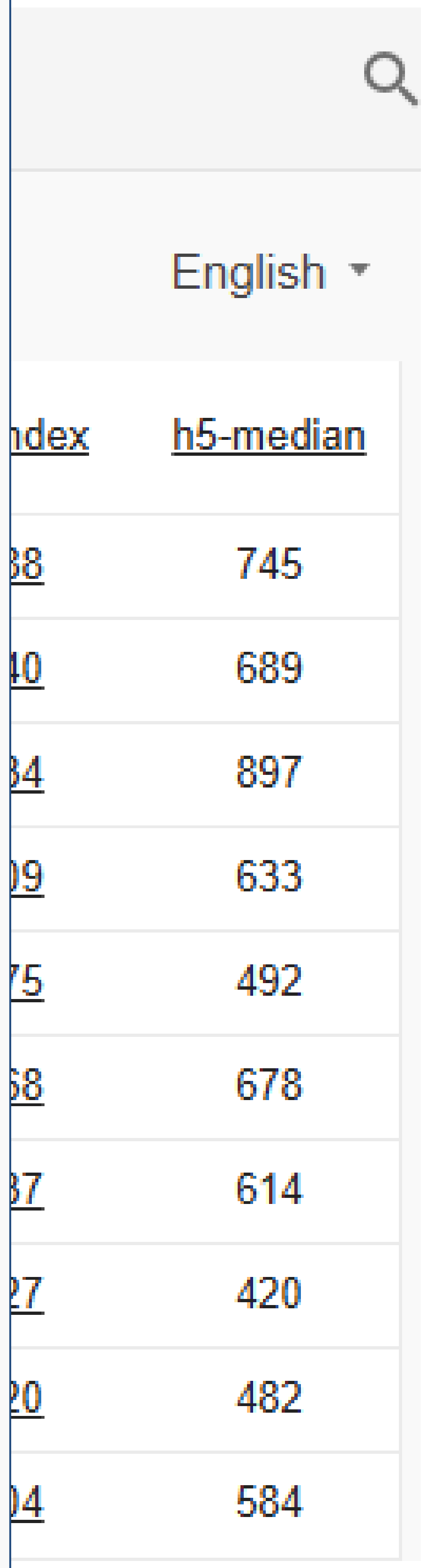


International Conference on Learning Representations

h5-index:304 h5-median:584

#2 Artificial Intelligence
#4 Engineering & Computer Science

Title / Author	Cited by
An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. A Dosovitskiy, L Beyer, A Kolesnikov, D Weissenborn, X Zhai, ... ICLR	38519
Decoupled Weight Decay Regularization. I Loshchilov, F Hutter ICLR (Poster)	18046
Measuring and Improving the Use of Graph Information in Graph Neural Networks. Y Hou, J Zhang, J Cheng, K Ma, RTB Ma, H Chen, MC Yang ICLR	7849
ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. Z Lan, M Chen, S Goodman, K Gimpel, P Sharma, R Soricut ICLR	7127
Large Scale GAN Training for High Fidelity Natural Image Synthesis. A Brock, J Donahue, K Simonyan ICLR	5675
DARTS: Differentiable Architecture Search. H Liu, K Simonyan, Y Yang ICLR (Poster)	4888
LoRA: Low-Rank Adaptation of Large Language Models. EJ Hu, Y Shen, P Wallis, Z Allen-Zhu, Y Li, S Wang, L Wang, W Chen ICLR	4881
Deformable DETR: Deformable Transformers for End-to-End Object Detection. X Zhu, W Su, L Lu, B Li, X Wang, J Dai ICLR	4444
BERTScore: Evaluating Text Generation with BERT. T Zhang, V Kishore, F Wu, KQ Weinberger, Y Artzi ICLR	4143
ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. K Clark, MT Luong, QV Le, CD Manning ICLR	3903



English ▾

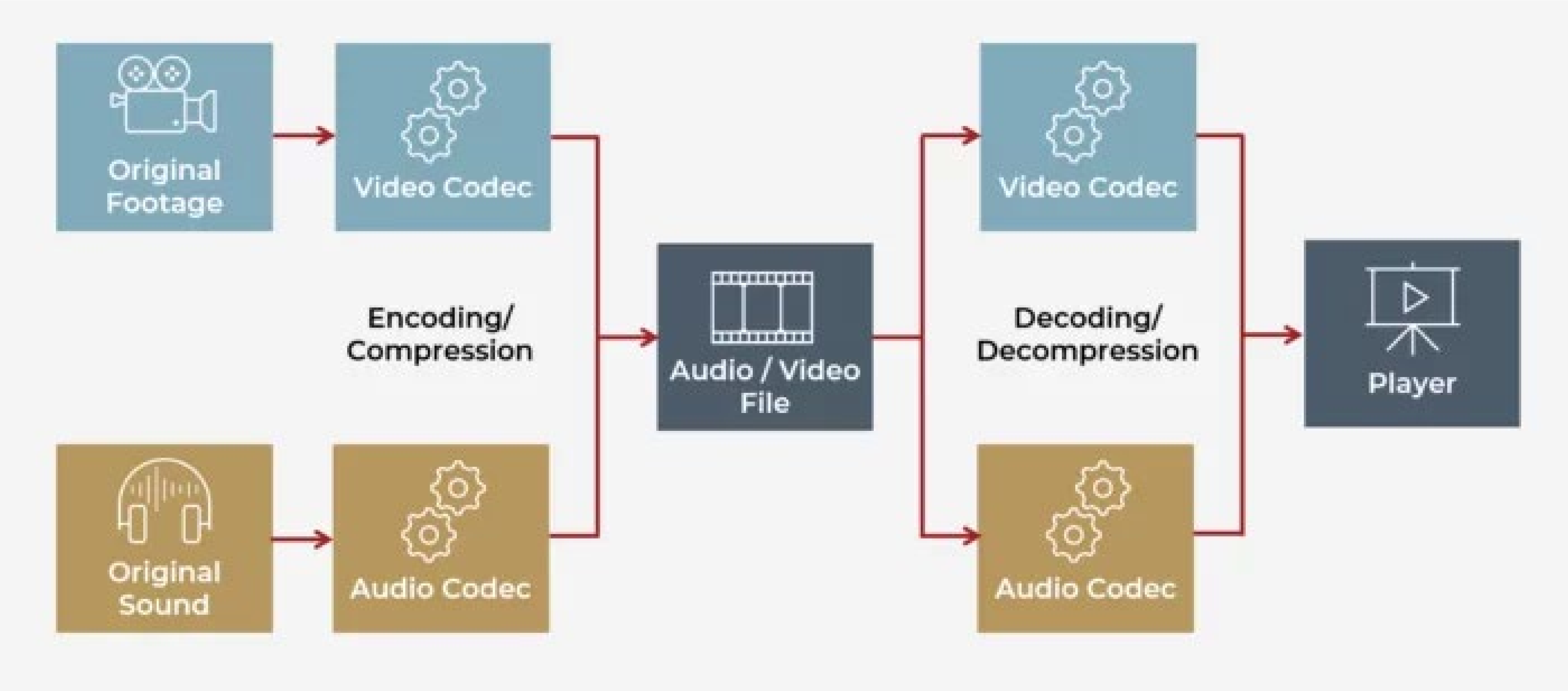
ndex	h5-median
38	745
40	689
34	897
09	633
75	492
58	678
37	614
27	420
20	482
04	584

Warning

I might use the terms “latent”, “embedding”, “representation”, “feature” interchangeably.

Motivation (kind of): Compression

- The idea is similar to compression (signal processing) or hashing (data structures):
  - ○ **encode** an image into a smaller vector s.t. you can **decode** it back to its original form
    - – Example: images, audio, video are stored in a compressed form on your computer using compression algorithms like JPEG, MP3, MPEG etc. The computer has software to decode it back so that you can view it (everytime you "open" a JPEG file to view an image, the decoder runs and converts code to RGB)



Motivation (kind of): Compression

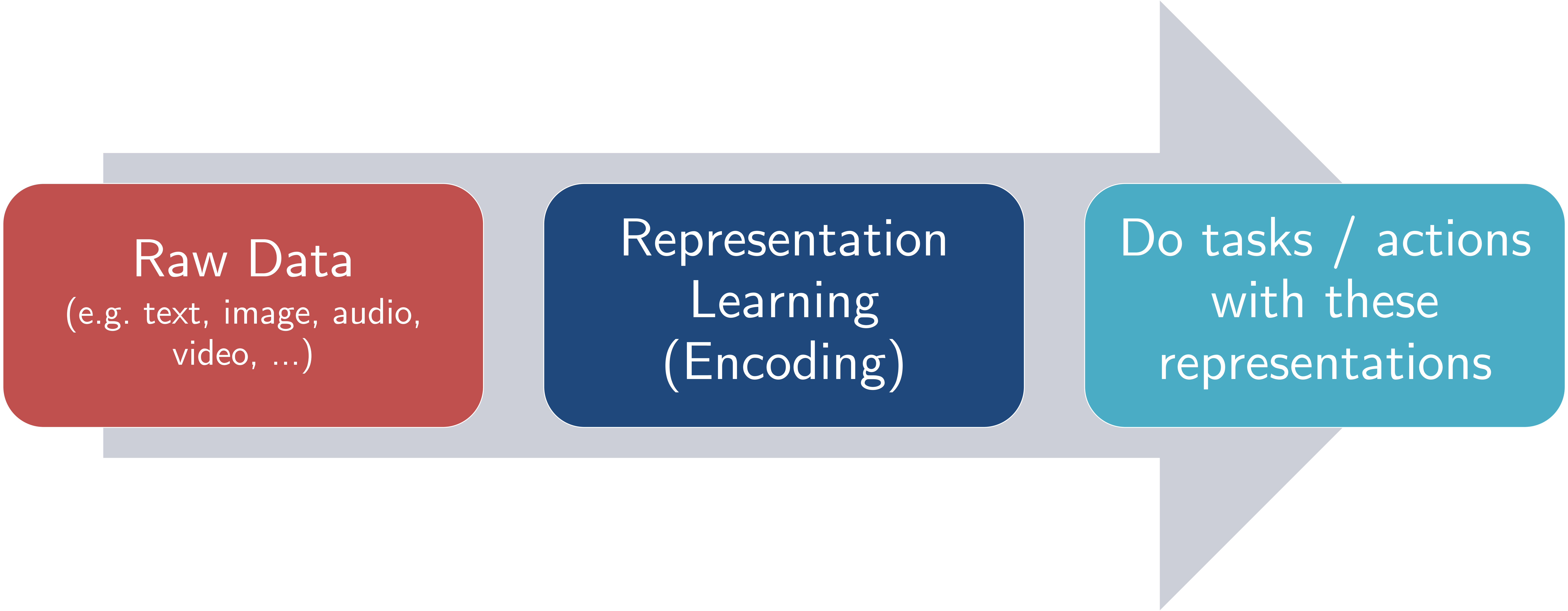
- The idea is similar to compression (signal processing) or hashing (information theory):
  - o encode an image into a smaller vector s.t. you can **decode** it back to its original form
    - – Example: images, audio, video are stored in a compressed form on your computer using compression algorithms like JPEG, MP3, MPEG etc. The computer has software to decode it back so that you can view it (everytime you "open" a JPEG file to view an image, the decoder runs and converts code to RGB)

- **Representation Learning**
  ~ convert inputs automatically into "codes" (called representations/embeddings / features) s.t. the representations are:
  - o Useful for downstream tasks (e.g. classification, regression, ...)
  - o "explain the data" and are "meaningful"

- Main difference: "meaningful" representation spaces to do "tasks"

  (The goal for compression is only efficient storage — not data classification/clustering etc.)

Representation Learning Paradigm



Raw Data
(e.g. text, image, audio, video, ...)

Representation Learning (Encoding)

Do tasks / actions with these representations

Typical Goals for Representations

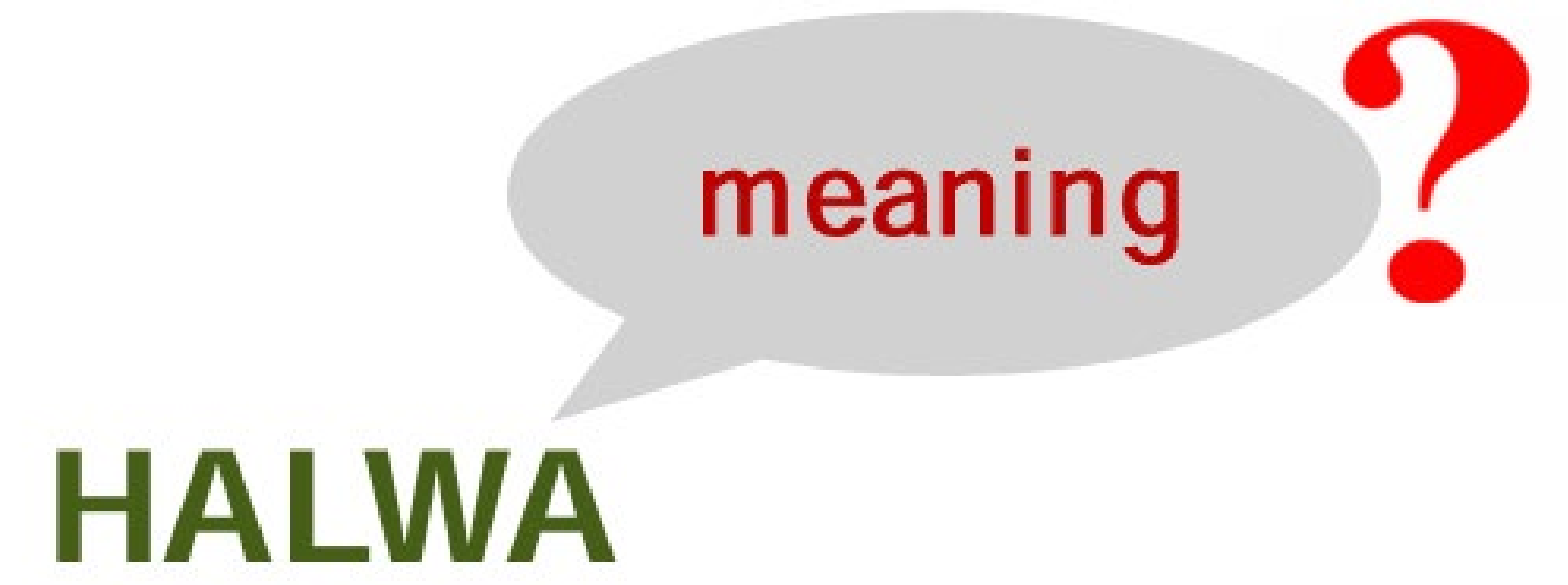
Similar representations for similar concepts

Food

Source: Pankaj Gupta (LMU Munich)

Typical Goals for Representations

A Semblance of “Context” should be encoded ...



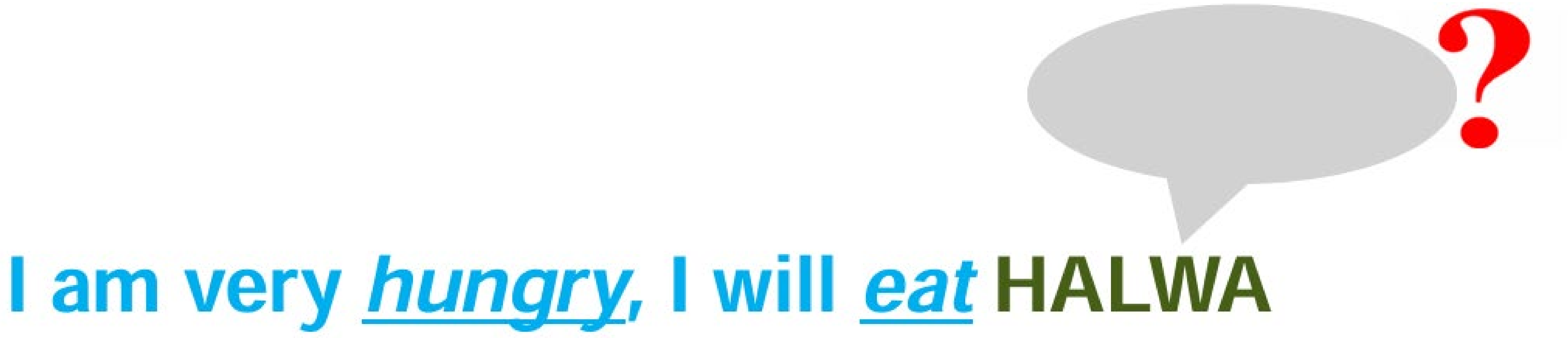
If you know the answer,
don’t share it with the class yet.

People from lands between Greece and India
might know the answer ...

Source: Pankaj Gupta (LMU Munich)

Typical Goals for Representations

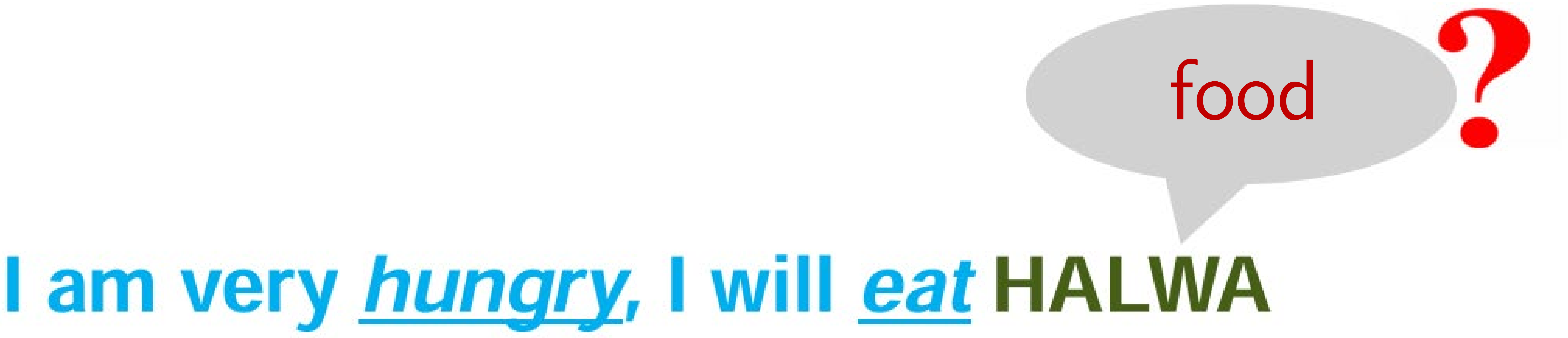
A Semblance of “Context” should be encoded ...



Source: Pankaj Gupta (LMU Munich)

Typical Goals for Representations

A Semblance of "Context" should be encoded ...



I am very *hungry*, I will *eat* HALWA

Typical Goals for Representations

A Semblance of "Context" should be encoded ...

I am very *hungry*, I will *eat* HALWA

If you speak Marathi, this word has two meanings depending on context

Halwa (1): a food item	*derived from: Farsi*
Halwa (2): (an instruction to) move (something)	*derived from: Sanskrit*

Typical Goals for Representations

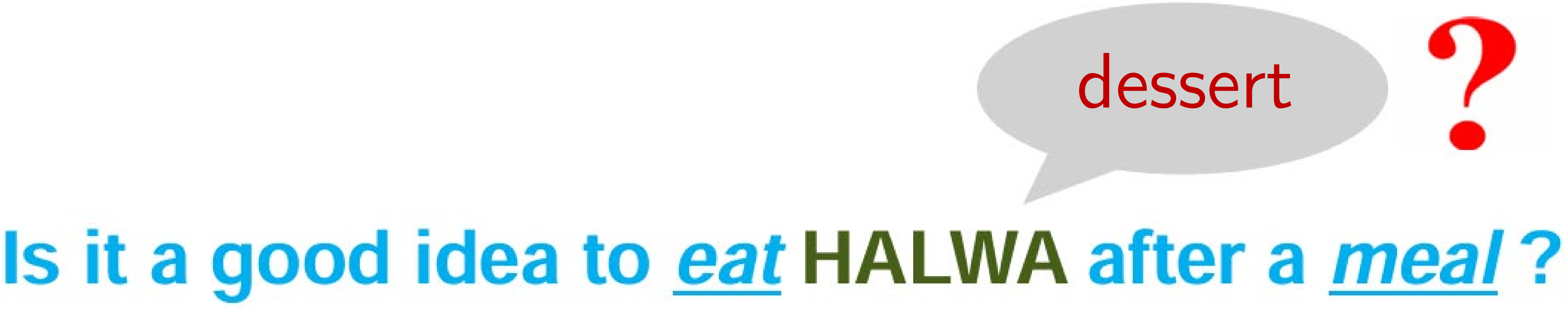
A Semblance of “Context” should be encoded ...

Is it a good idea to *eat* HALWA after a *meal* ?

Source: Pankaj Gupta (LMU Munich)

Typical Goals for Representations

A Semblance of “Context” should be encoded ...



Is it a good idea to *eat* HALWA after a *meal* ?

Source: Pankaj Gupta (LMU Munich)

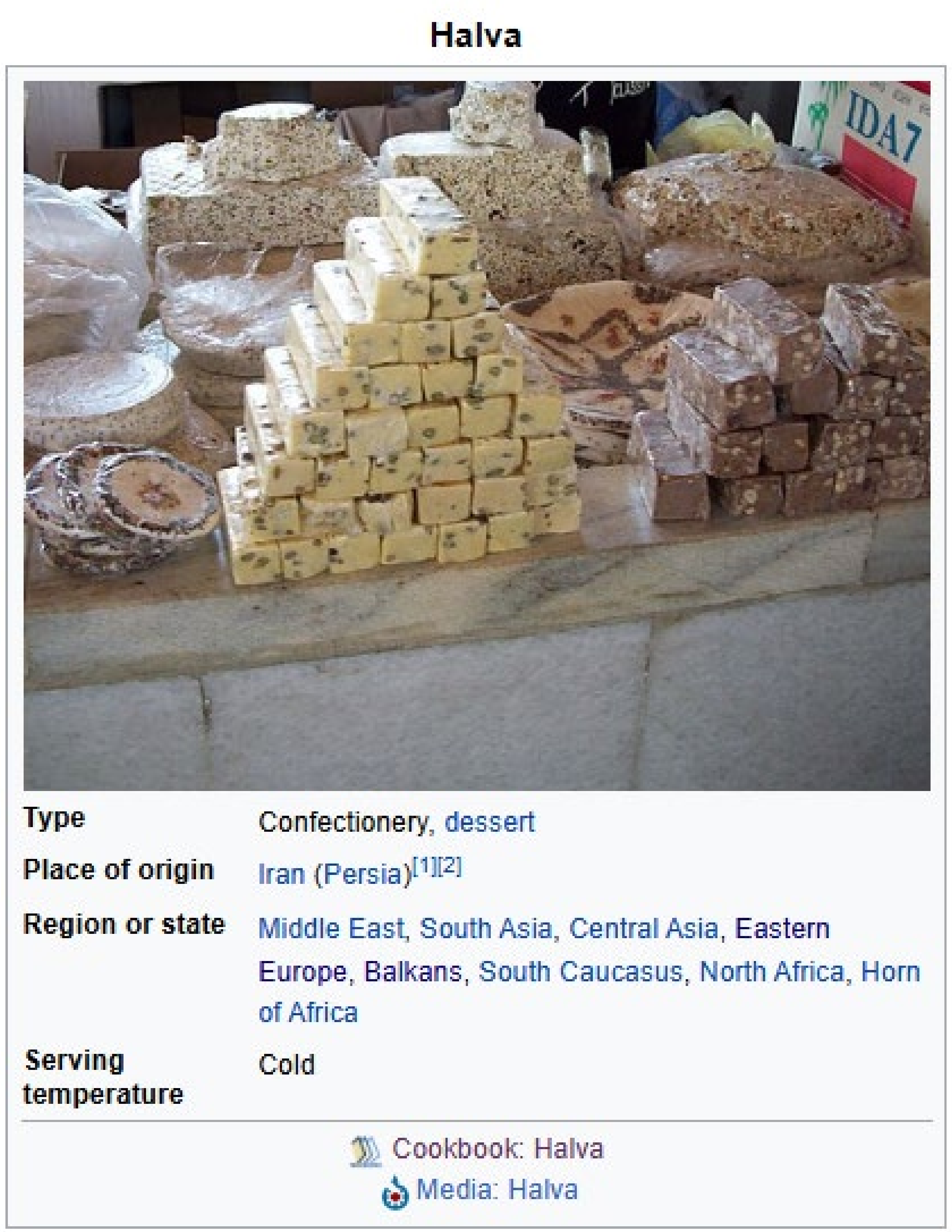
Typical Goals for Representations

A Semblance of “Context” should be encoded ...

sugary dessert ?

Oh no! I forgot to put sugar in the HALWA

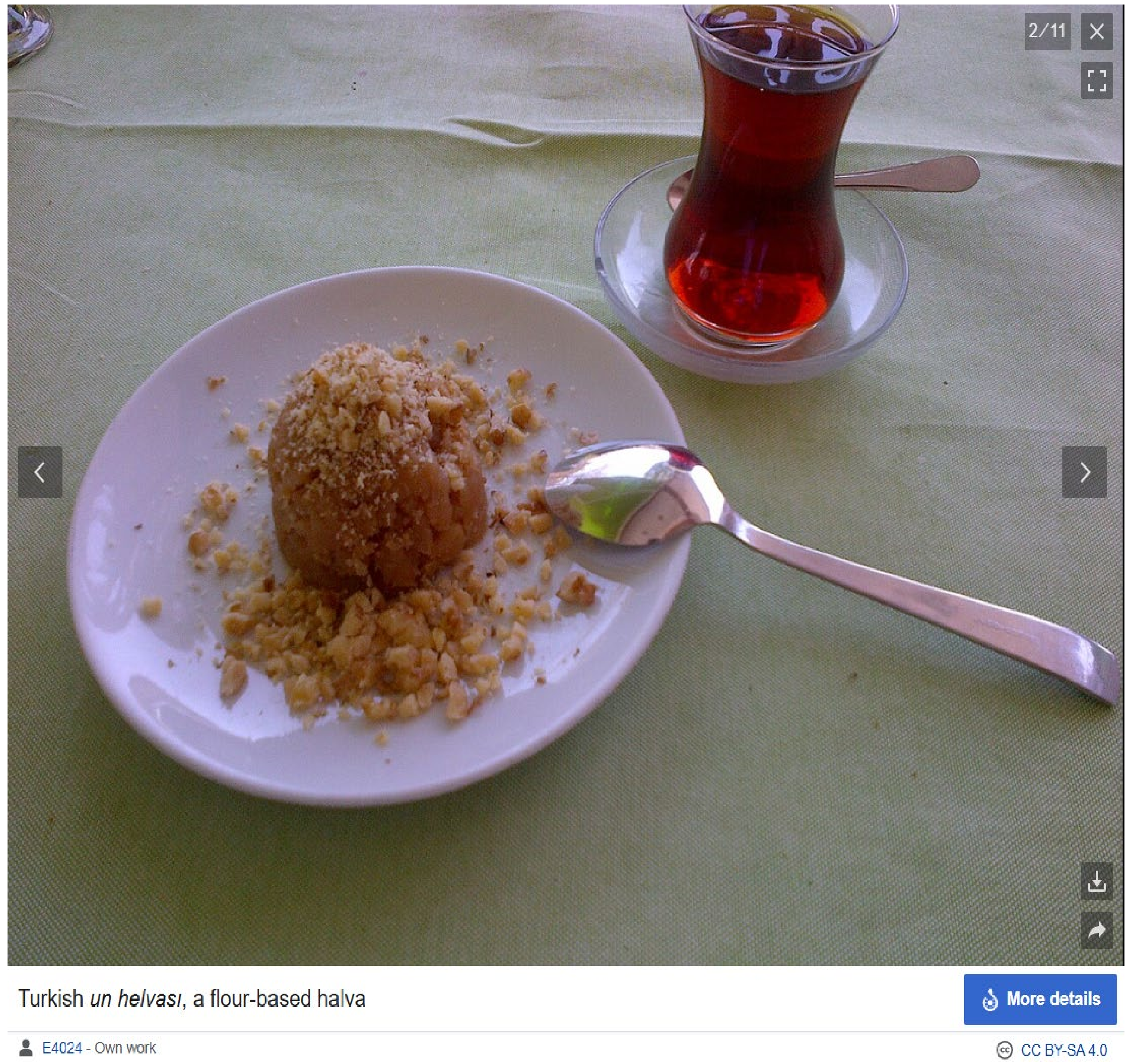
Source: Pankaj Gupta (LMU Munich)



Halva

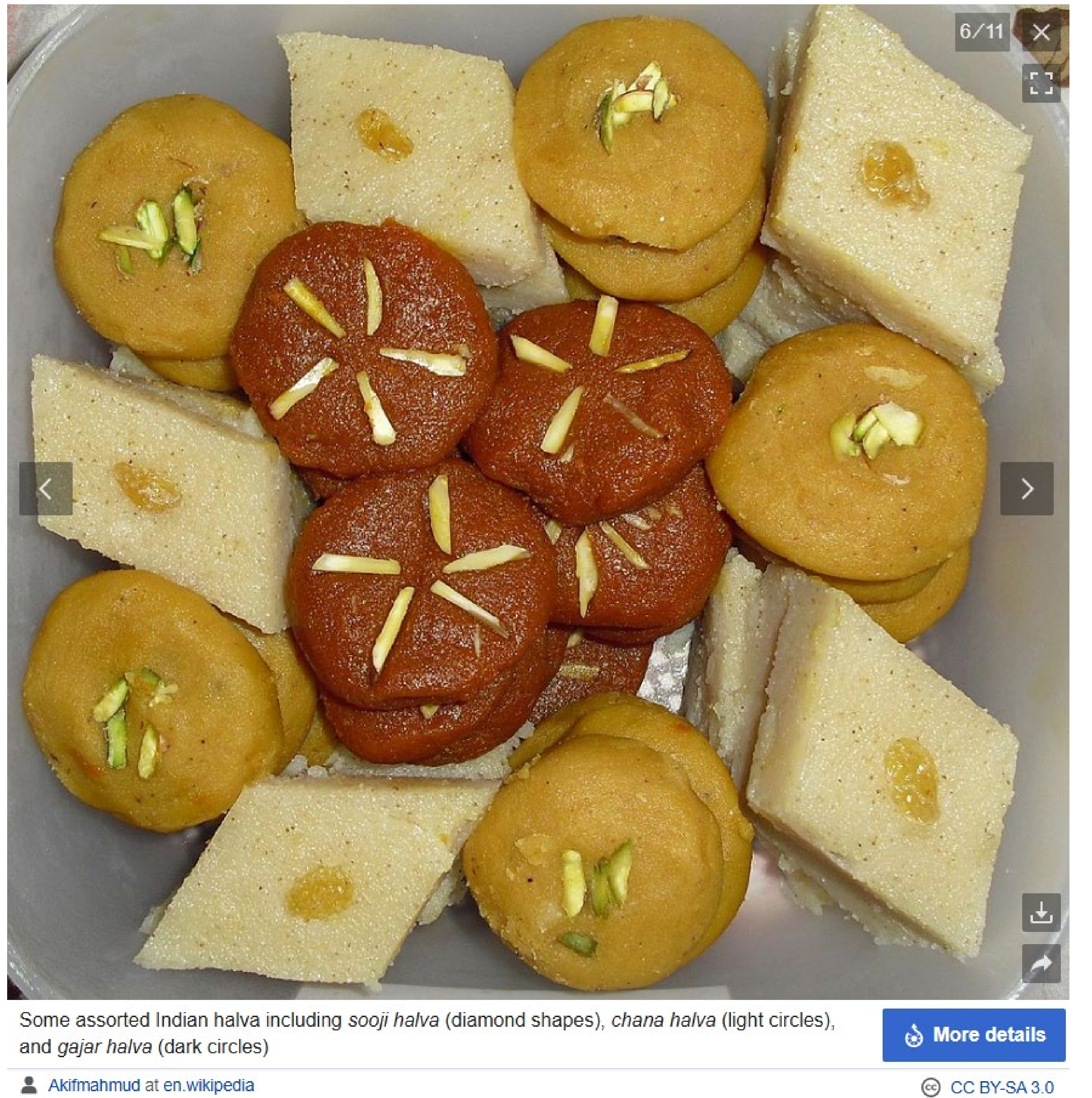
Type	Confectionery, dessert
Place of origin	Iran (Persia)[1][2]
Region or state	Middle East, South Asia, Central Asia, Eastern Europe, Balkans, South Caucasus, North Africa, Horn of Africa
Serving temperature	Cold

Cookbook: Halva
Media: Halva



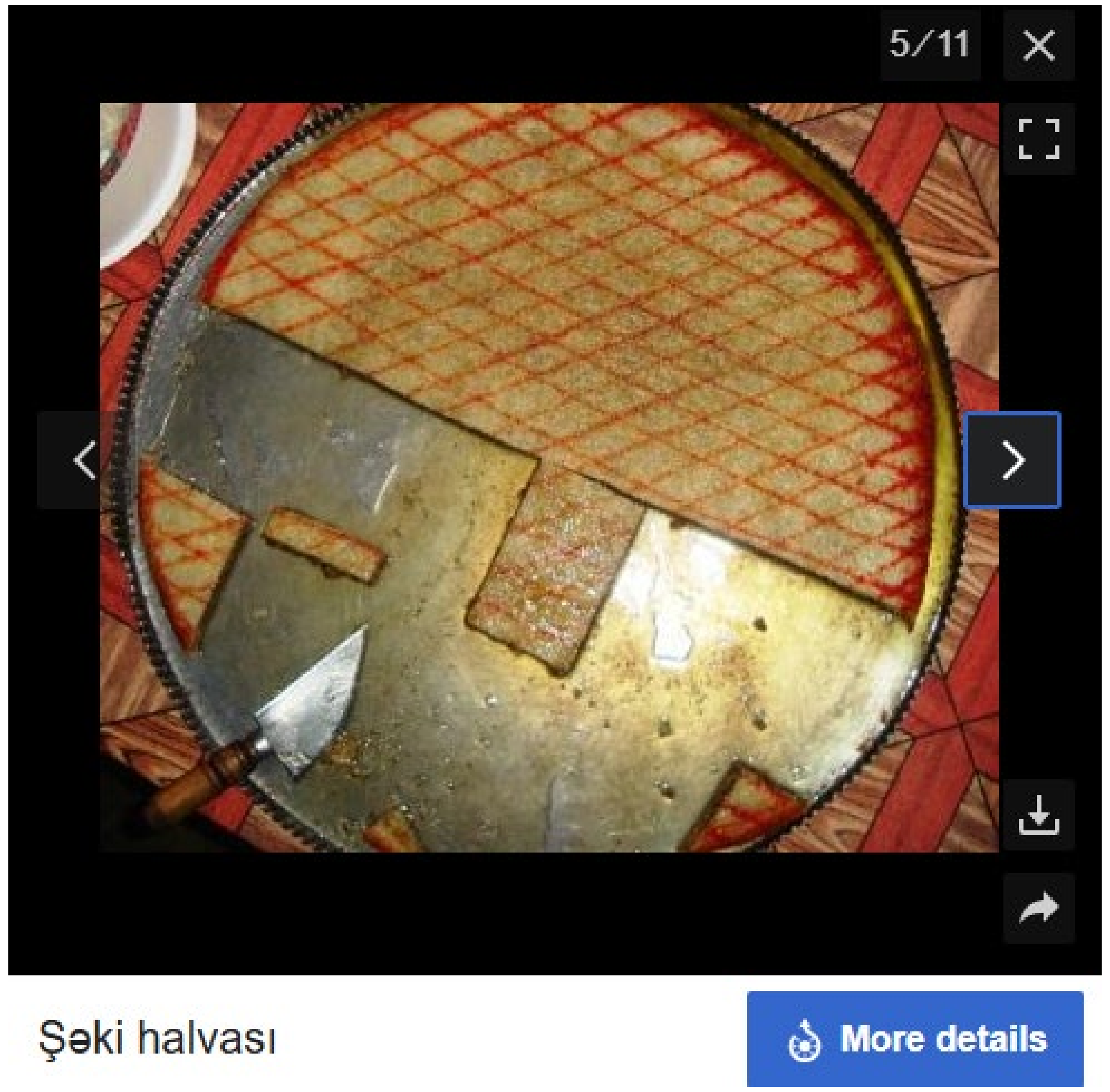
Turkish *un helvası*, a flour-based halva

E4024 - Own work CC BY-SA 4.0

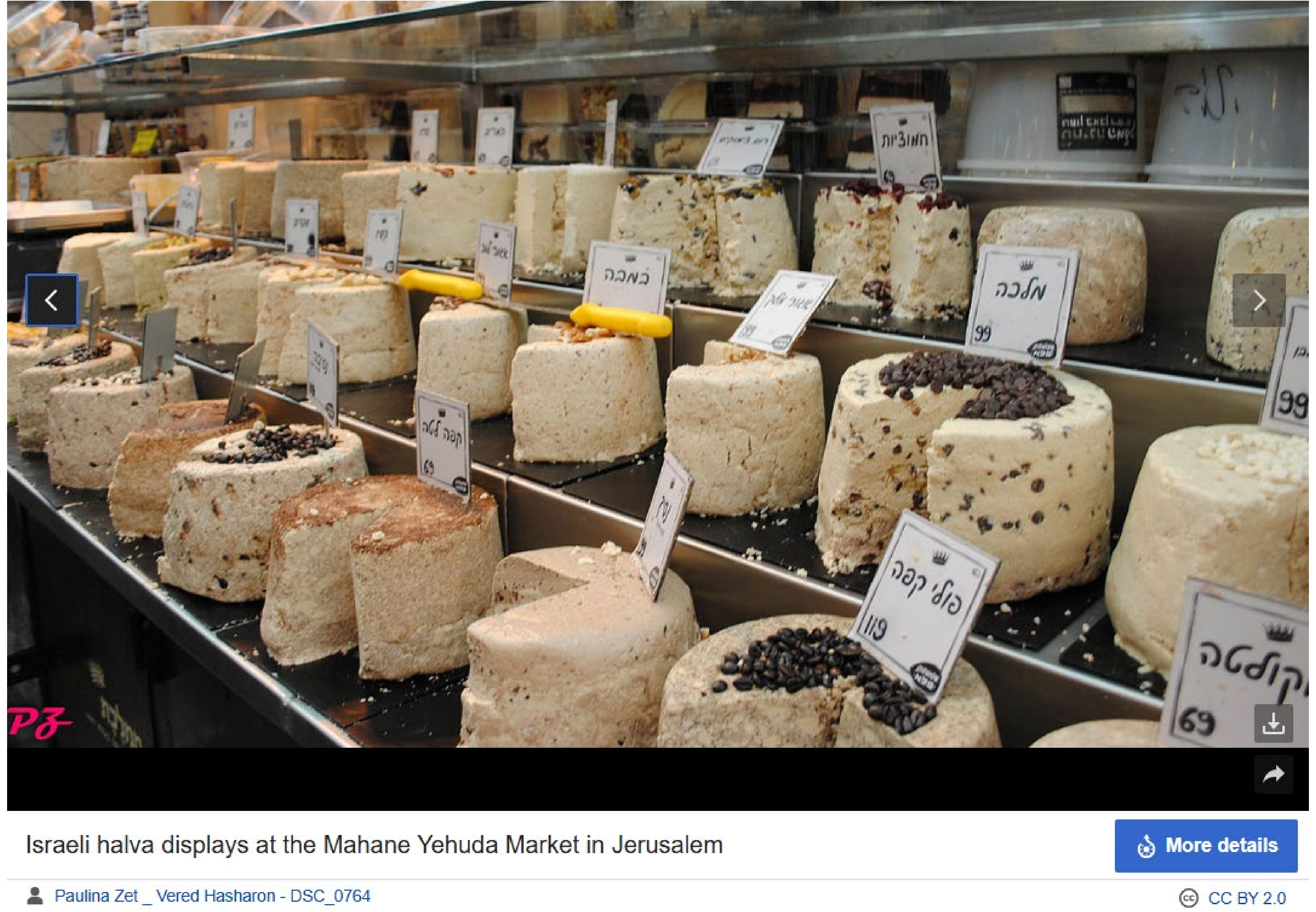


Some assorted Indian halva including *sooji halva* (diamond shapes), *chana halva* (light circles), and *gajar halva* (dark circles)

Akifmahmud at en.wikipedia CC BY-SA 3.0



Şəki halvası



Israeli halva displays at the Mahane Yehuda Market in Jerusalem

Paulina Zet _ Vered Hasharon - DSC_0764 CC BY 2.0

Typical Goals for Representations

Parts, properties, attributes, ontology ?

- “bird” has “wing”, “beak”, “feathers”
- “bird” can “fly”
- “bird” is under category “animal”
- “bird” has subcategories “eagle”, “peacock”, “sparrow”, “seagull”, “pigeon”

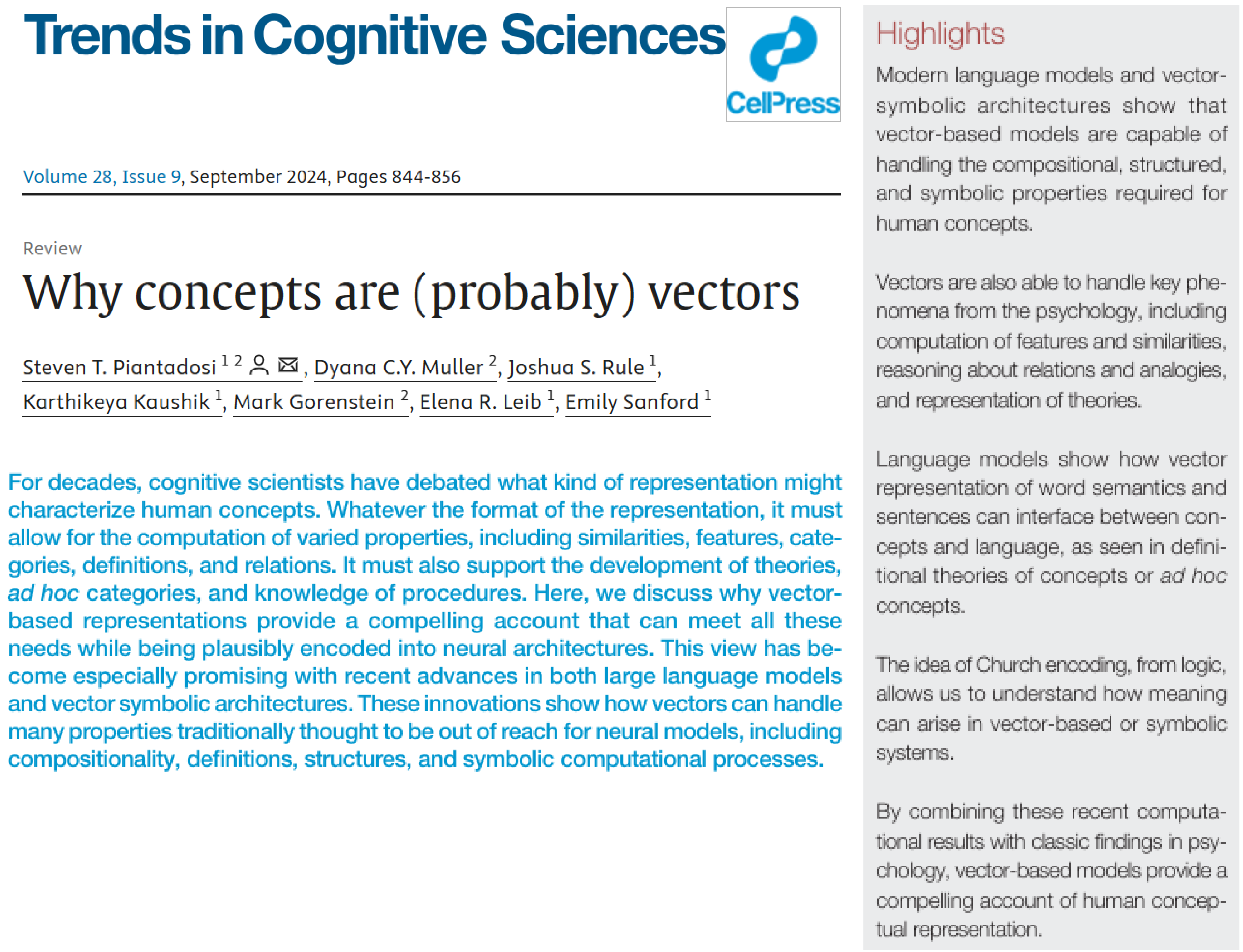
Source: Pankaj Gupta (LMU Munich)

Representation Learning is a Philosophy for Learning

Key assumptions in this philosophy:

- You can convert a high-dimensional input space into a low-dimensional representation space
  - Example: RGB images → 100 dim vectors
- A good representation space will have a "structure"
  - Example: Similarity, Symmetry, Relations will be easy to understand
  - Why? So that we can do arithmetic in representation space to do tasks
- Representations can be learned from data
- Representations can be leveraged for doing tasks

Parallel Work in Cog.Sci.



Trends in Cognitive Sciences

Volume 28, Issue 9, September 2024, Pages 844-856

Review

Why concepts are (probably) vectors

Steven T. Piantadosi [1] [2], Dyana C.Y. Muller [2], Joshua S. Rule [1], Karthikeya Kaushik [1], Mark Gorenstein [2], Elena R. Leib [1], Emily Sanford [1]

For decades, cognitive scientists have debated what kind of representation might characterize human concepts. Whatever the format of the representation, it must allow for the computation of varied properties, including similarities, features, categories, definitions, and relations. It must also support the development of theories, *ad hoc* categories, and knowledge of procedures. Here, we discuss why vector-based representations provide a compelling account that can meet all these needs while being plausibly encoded into neural architectures. This view has become especially promising with recent advances in both large language models and vector symbolic architectures. These innovations show how vectors can handle many properties traditionally thought to be out of reach for neural models, including compositionality, definitions, structures, and symbolic computational processes.

Highlights

Modern language models and vector-symbolic architectures show that vector-based models are capable of handling the compositional, structured, and symbolic properties required for human concepts.

Vectors are also able to handle key phenomena from the psychology, including computation of features and similarities, reasoning about relations and analogies, and representation of theories.

Language models show how vector representation of word semantics and sentences can interface between concepts and language, as seen in definitional theories of concepts or *ad hoc* concepts.

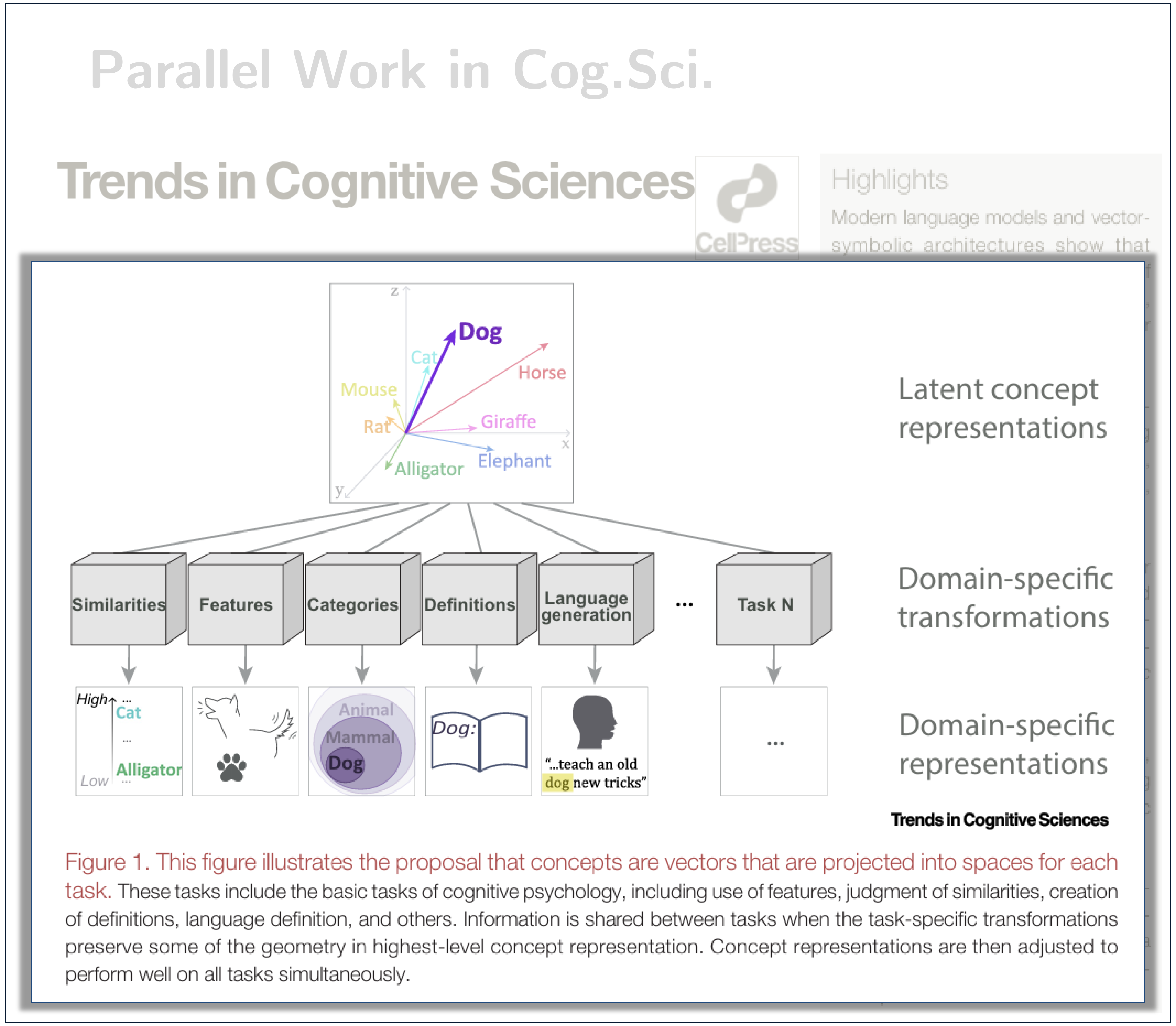
The idea of Church encoding, from logic, allows us to understand how meaning can arise in vector-based or symbolic systems.

By combining these recent computational results with classic findings in psychology, vector-based models provide a compelling account of human conceptual representation.

Representation Learning is a Philosophy for Learning

Key assumptions in this philosophy:

- You can convert a high-dimensional input space into a low-dimensional representation space
    - Example: RGB images → 100 dim vectors
- A good representation space will have a "structure"
    - Example: Similarity, Symmetry, Relations will be easy to understand
    - Why? So that we can do arithmetic in representation space to do tasks
- Representations can be learned from data
- Representations can be leveraged for doing tasks



Parallel Work in Cog.Sci.

Trends in Cognitive Sciences

Highlights

Modern language models and vector-symbolic architectures show that

Figure 1. This figure illustrates the proposal that concepts are vectors that are projected into spaces for each task. These tasks include the basic tasks of cognitive psychology, including use of features, judgment of similarities, creation of definitions, language definition, and others. Information is shared between tasks when the task-specific transformations preserve some of the geometry in highest-level concept representation. Concept representations are then adjusted to perform well on all tasks simultaneously.

Tell us how it works ...

Types of Modeling (Probabilistic Interpretation)

Data: x; **Label: y**

"cat"

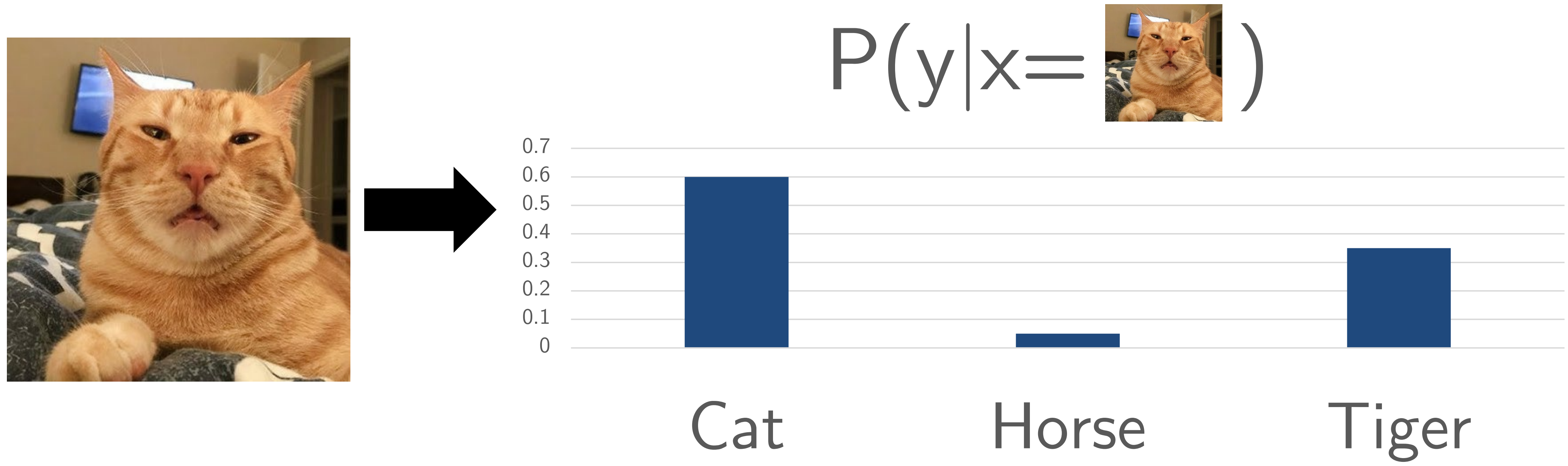
Density Function: $p(x)$

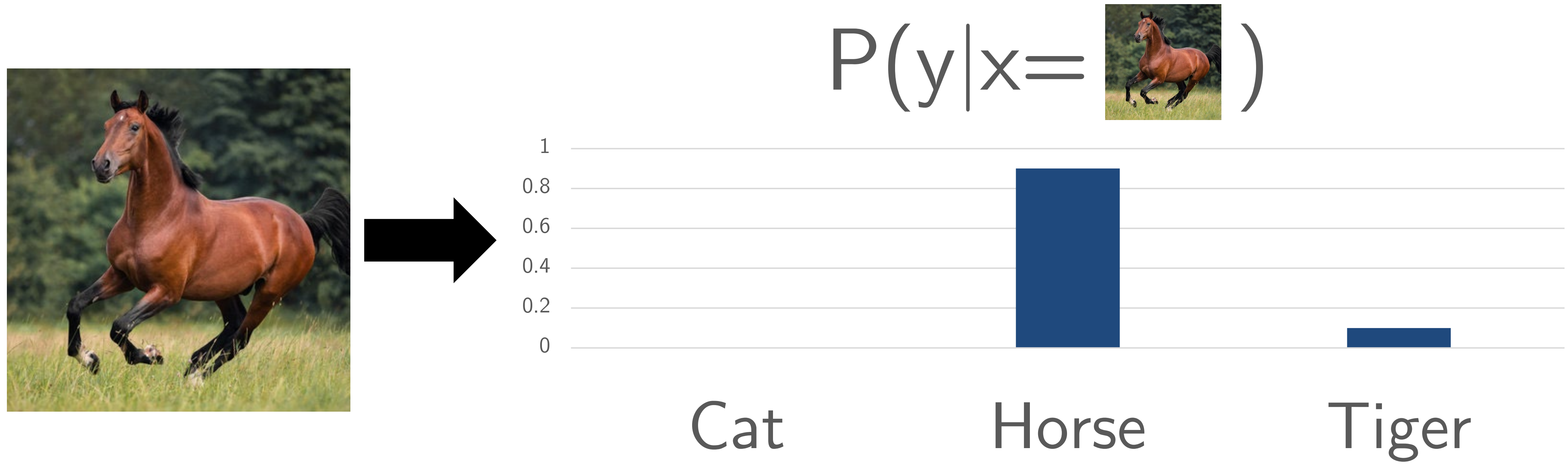
$$\int_X p(x)\,dx = 1$$

(probabilities of all inputs sum to 1)

Discriminative Model

Learn Prob. Dist. $P(y|x)$





$$\forall x, \sum_c P(y=c|x) = 1$$

Types of Modeling (Probabilistic Interpretation)

Data: x; **Label: y**

"cat"

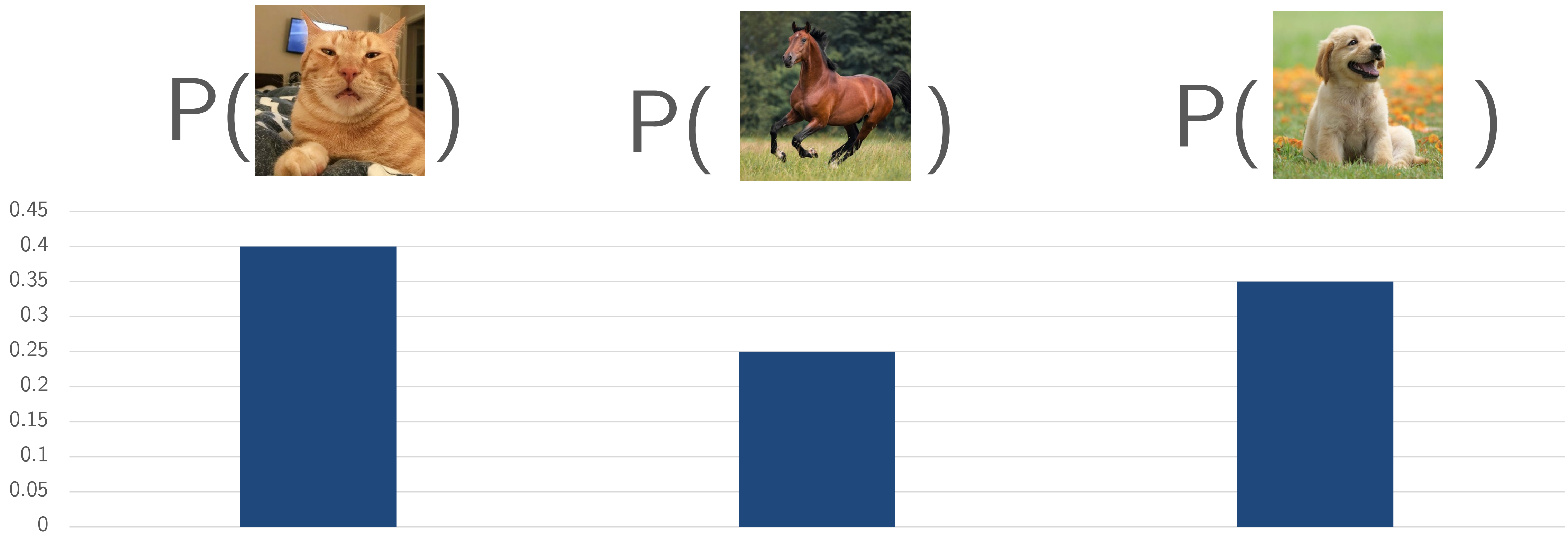
Density Function: $p(x)$

$$\int_X p(x)\,dx = 1$$

(probabilities of all inputs sum to 1)

Generative Model

Learn Marginal Prob. Dist. $P(x)$



Conditional Generative Model

Learn conditional probability $P(x|y)$

$$\underbrace{\boxed{P(x\mid y)}}_{\text{Conditional Generative Model}} = \frac{\overbrace{\boxed{P(y\mid x)}}^{\text{Discriminative Model}}}{\underbrace{\boxed{P(y)}}_{\text{Prior over labels}}}\overbrace{\boxed{P(x)}}^{\text{(Unconditional) Generative Model}}$$

Types of Modeling (Probabilistic Interpretation)

Data: x; **Label: y**

"cat"

Density Function: $p(x)$

$\int_X p(x)\,dx = 1$

(probabilities of all inputs sum to 1)

- **Discriminative Model**

  Learn Prob. Dist. $P(y|x)$

- **Generative Model**

  Learn Marginal Prob. Dist. $P(x)$

- **Conditional Generative Model**

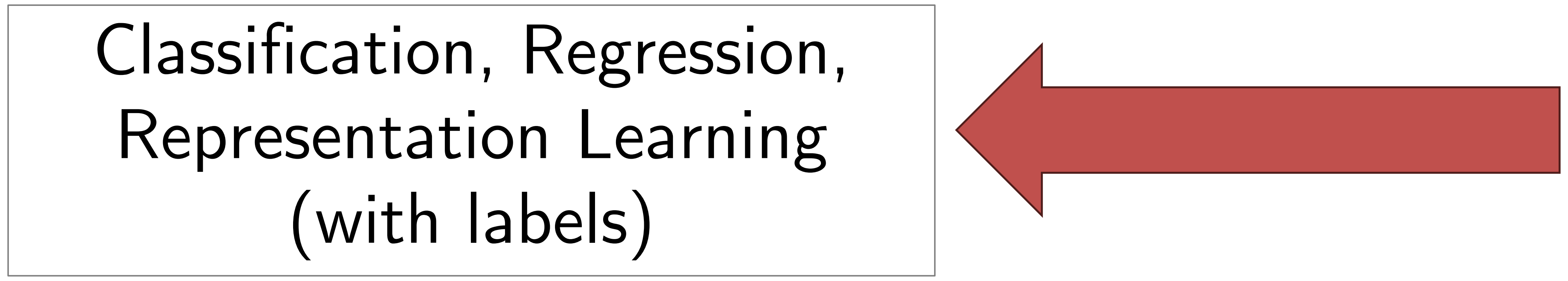
  Learn conditional probability $P(x|y)$

Types of Modeling (Probabilistic Interpretation)

APPLICATIONS

- **Discriminative Model**

Learn Prob. Dist. $P(y|x)$



Classification, Regression, Representation Learning (with labels)

- **Generative Model**

Learn Marginal Prob. Dist. $P(x)$

- **Conditional Generative Model**

Learn conditional probability $P(x|y)$

Types of Modeling (Probabilistic Interpretation)

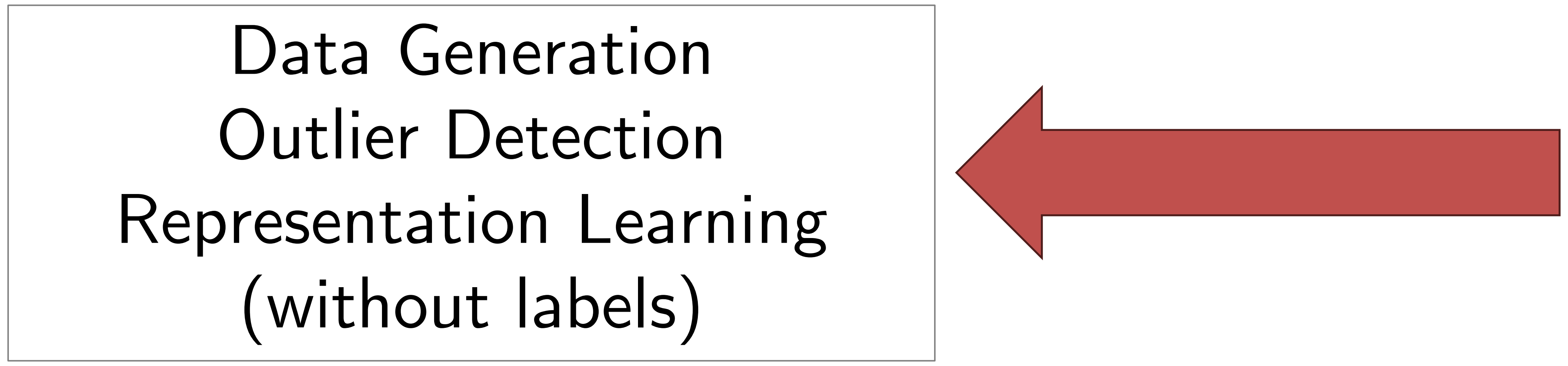
APPLICATIONS

- **Discriminative Model**

  Learn Prob. Dist. $P(y|x)$

- **Generative Model**

  Learn Marginal Prob. Dist. $P(x)$



Data Generation
Outlier Detection
Representation Learning
(without labels)

- **Conditional Generative Model**

  Learn conditional probability $P(x|y)$

Types of Modeling (Probabilistic Interpretation)

APPLICATIONS

- **Discriminative Model**

  Learn Prob. Dist. $P(y|x)$

- **Generative Model**

  Learn Marginal Prob. Dist. $P(x)$

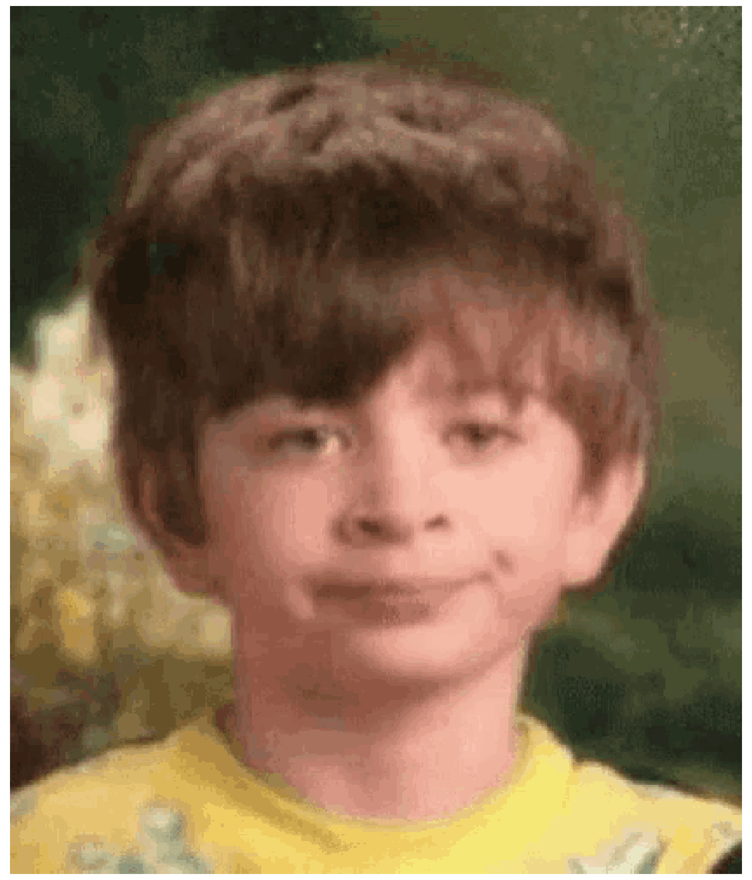
- **Conditional Generative Model**

  Learn conditional probability $P(x|y)$

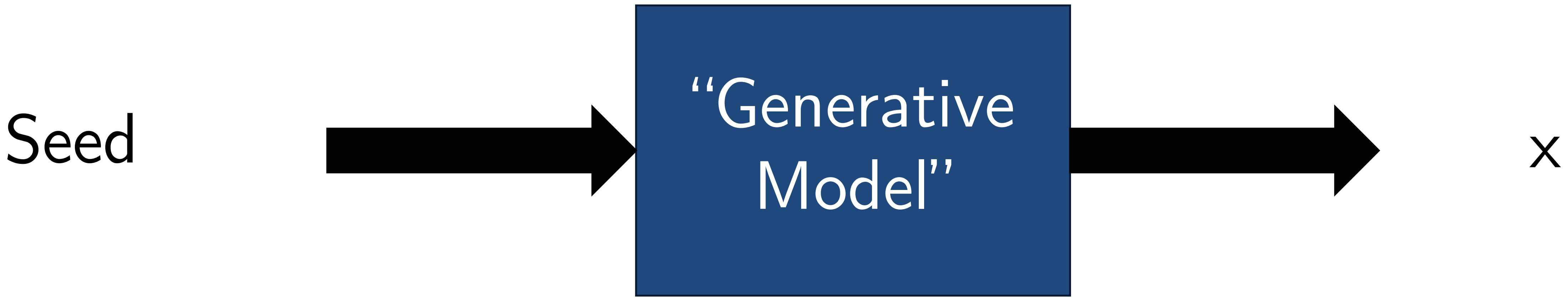
Machine Translation
Text-to-image generation

(pretty much every "GenAI" product you see is a conditional generative model)

Generative Models



- What's a Generative Model?
  - A model for the probability distribution of data x P(x)
  - A model that can be used to "generate" data marketing term "genAI"



- Generative Models can be *learned*
  - *You are given some observed data X* *(e.g. face images)*
  - *You choose a function (e.g. neural network) to model $P(x;\theta)$ using parameters θ*
  - *You estimate θ s.t. $P(x;\theta)$ best fits the observations X*

Generative Models

- Generative Models can be *learned*
  - *You estimate* θ *s.t.* $P(x;\theta)$ **best fits** *the observations X*

- '**Best fit"** in what sense?
  - *Maximum Likelihood* $\theta^* = \operatorname*{argmax}_\theta P(x;\theta)$

- How to model the distribution of high dimensional data?

$$P(x) = \int_z P(x,z)\,dz = \int_z P(z)P(x|z)$$

  - $P_\theta(z)$ and $P_\theta(x|z)$ can be factorized

$$P_\theta(x|z) = P_\theta(x^1 \dots, x^D\ |z) = \prod_i P_\theta(x^i|z)$$

$$\theta^* = \operatorname*{argmax}_\theta P_\theta(x) = \operatorname*{argmax}_\theta \prod_i P_\theta(x^i|z) = \operatorname*{argmax}_\theta \log \sum_i P_\theta(x^i|z)$$