Representation of Information
ECE 598 LV – Lecture 10

Lav R. Varshney

20 February 2024

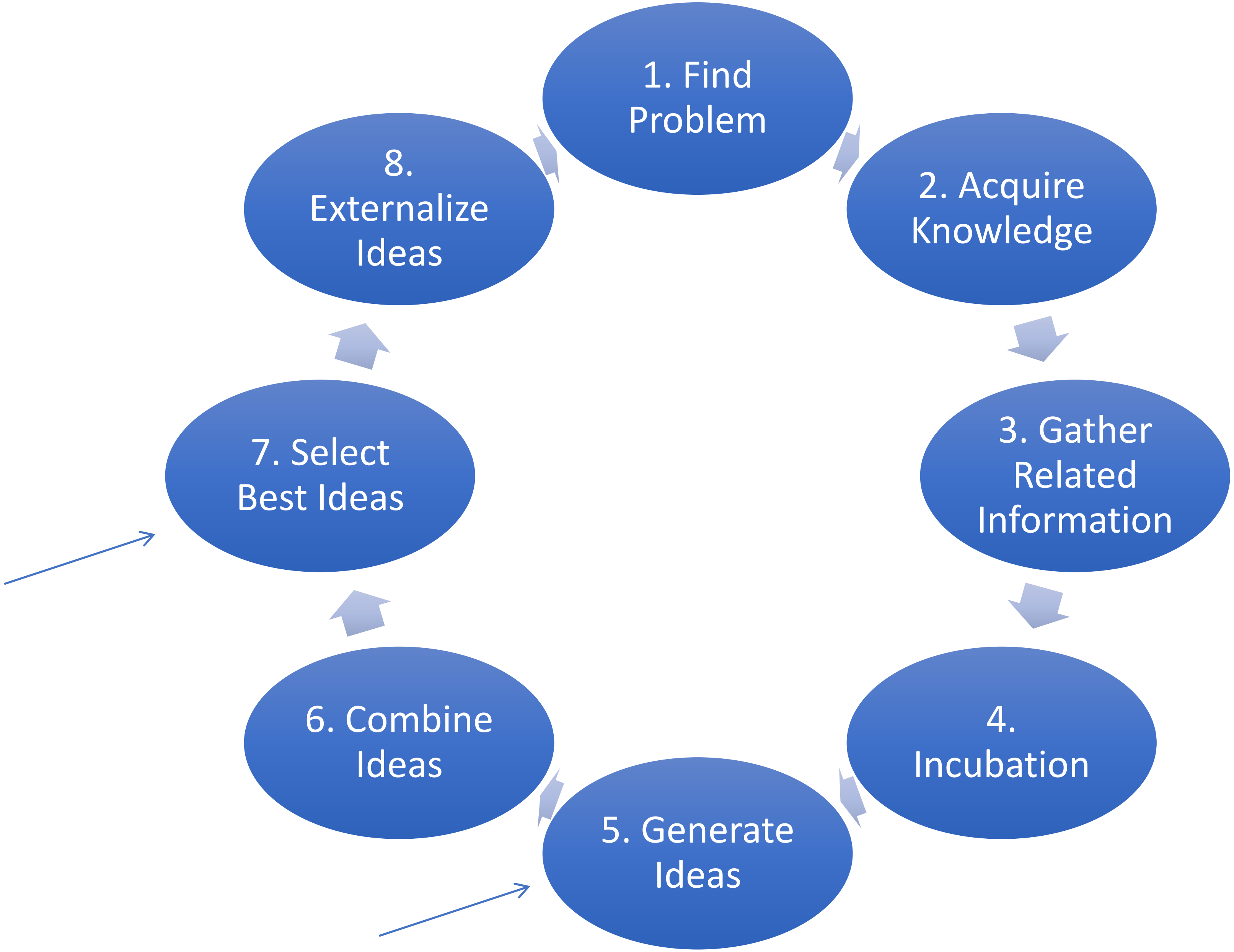
5:59 AM / JUNE 30, 2014

Former IBM Research scientist Lav Varshney presents a demo of an early version of the cognitive cooking technology at IBM Research.

CREDIT: COURTESY IBM

How IBM's Chef Watson Actually Works

WRITTEN BY ROCHELLE BILOW


1. Find Problem
2. Acquire Knowledge
3. Gather Related Information
4. Incubation
5. Generate Ideas
6. Combine Ideas
7. Select Best Ideas
8. Externalize Ideas

[Sawyer, 2012]

bon appétit | RESTAURANTS + TRAVEL | ENTERTAINING + STYLE | DRINKS | PEOPLE | EVENTS | TEST KITCHEN | VIDEO | RECIPES | SUBSCRIBE

BON APPÉTIT / ENTERTAINING + STYLE / TRENDS + NEWS 5:59 AM / JUNE 30, 2014

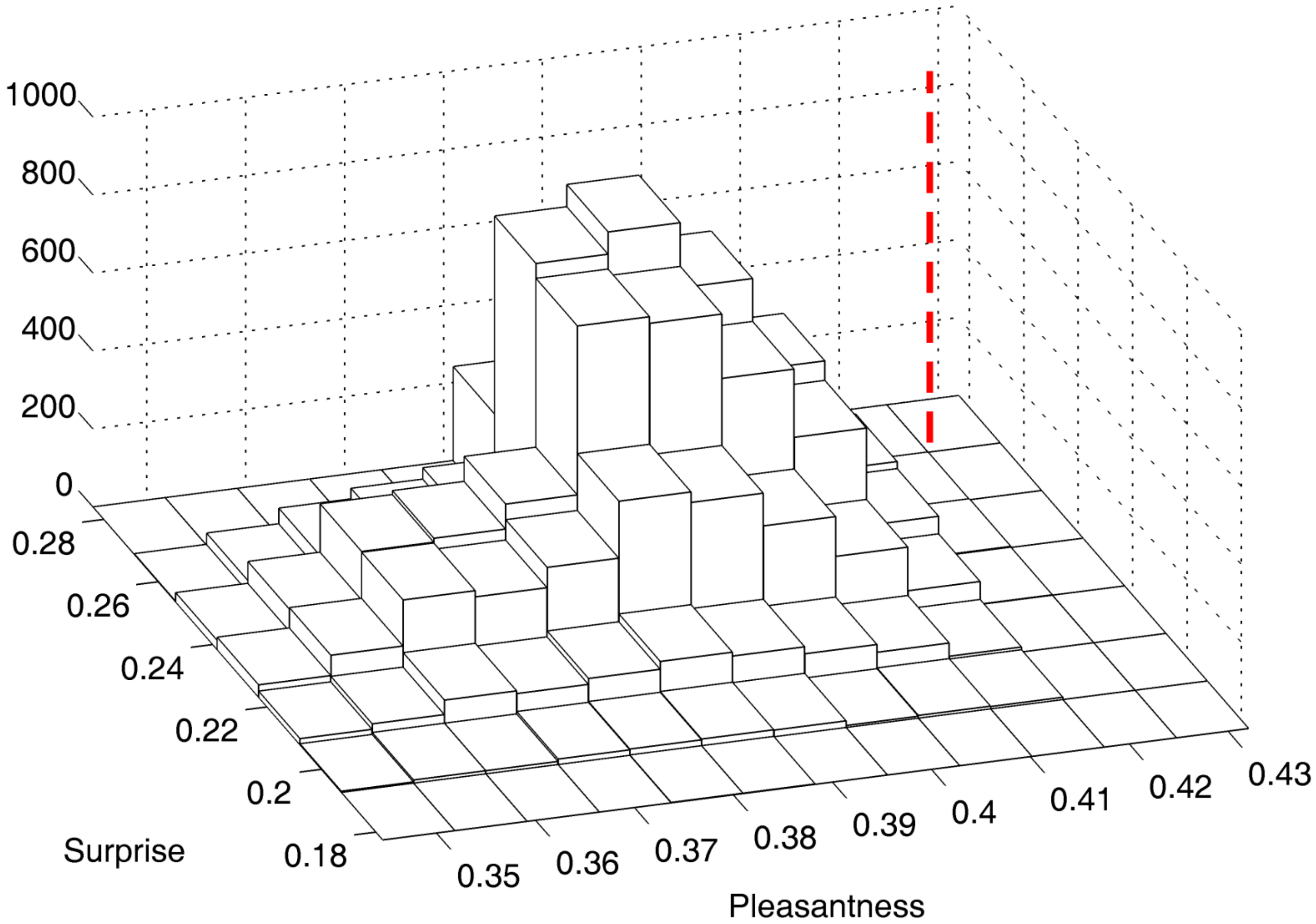
Former IBM Research scientist Lav Varshney presents a demo of an early version of the cognitive cooking technology at IBM Research. CREDIT: COURTESY IBM

How IBM's Chef Watson Actually Works

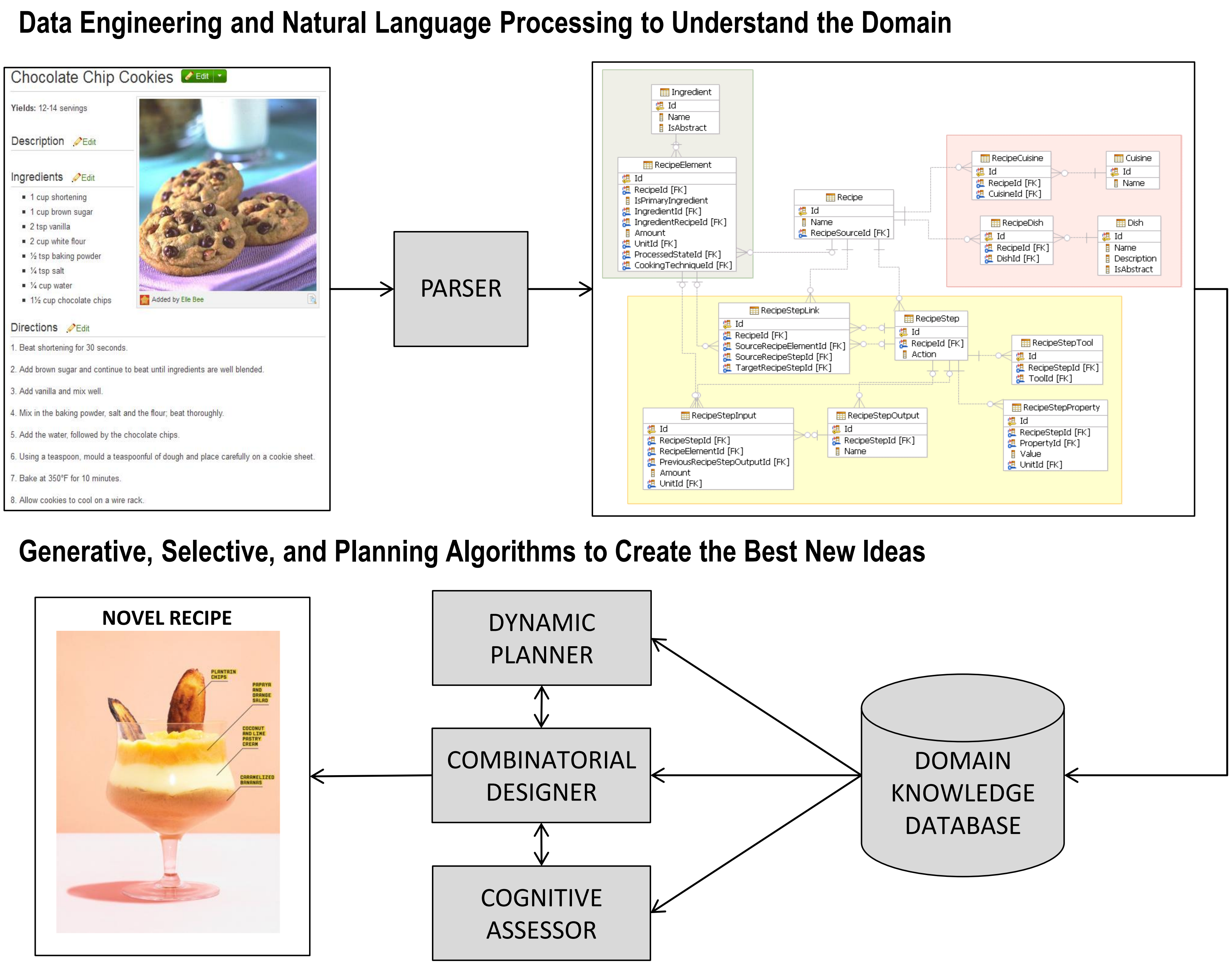
47 Comment

WRITTEN BY ROCHELLE BILOW PRINT RSS

1. Sample from state space, using culturally well-chosen sampling distribution
2. Rank according to psychophysical predictors of novelty and flavor
3. Select either automatically or semi-automatically depending on human-computer interaction model



Joint histogram of surprise and pleasantness for 10000 generated Caymanian Plantain Dessert recipes. Values for the selected/tested recipe indicated with red dashed line.


Data Engineering and Natural Language Processing to Understand the Domain
Chocolate Chip Cookies
Edit
Yields: 12-14 servings
Description
Edit
Ingredients
Edit
1 cup shortening
1 cup brown sugar
2 tsp vanilla
2 cup white flour
½ tsp baking powder
¼ tsp salt
¼ cup water
1½ cup chocolate chips
Added by Elle Bee
Directions
Edit
1. Beat shortening for 30 seconds.
2. Add brown sugar and continue to beat until ingredients are well blended.
3. Add vanilla and mix well.
4. Mix in the baking powder, salt and the flour; beat thoroughly.
5. Add the water, followed by the chocolate chips.
6. Using a teaspoon, mould a teaspoonful of dough and place carefully on a cookie sheet.
7. Bake at 350°F for 10 minutes.
8. Allow cookies to cool on a wire rack.
PARSER
Ingredient
Id
Name
IsAbstract
RecipeElement
Id
RecipeId [FK]
IsPrimaryIngredient
IngredientId [FK]
IngredientRecipeId [FK]
Amount
UnitId [FK]
ProcessedStateId [FK]
CookingTechniqueId [FK]
Recipe
Id
Name
RecipeSourceId [FK]
RecipeCuisine
Id
RecipeId [FK]
CuisineId [FK]
Cuisine
Id
Name
RecipeDish
Id
RecipeId [FK]
DishId [FK]
Dish
Id
Name
Description
IsAbstract
RecipeStepLink
Id
RecipeId [FK]
SourceRecipeElementId [FK]
SourceRecipeStepId [FK]
TargetRecipeStepId [FK]
RecipeStep
Id
RecipeId [FK]
Action
RecipeStepTool
Id
RecipeStepId [FK]
ToolId [FK]
RecipeStepInput
Id
RecipeStepId [FK]
RecipeElementId [FK]
PreviousRecipeStepOutputId [FK]
Amount
UnitId [FK]
RecipeStepOutput
Id
RecipeStepId [FK]
Name
RecipeStepProperty
Id
RecipeStepId [FK]
PropertyId [FK]
Value
UnitId [FK]
Generative, Selective, and Planning Algorithms to Create the Best New Ideas
NOVEL RECIPE
PLANTAIN CHIPS
PAPAYA AND ORANGE SALAD
COCONUT AND LIME PASTRY CREAM
CARAMELIZED BANANAS
DYNAMIC PLANNER
COMBINATORIAL DESIGNER
COGNITIVE ASSESSOR
DOMAIN KNOWLEDGE DATABASE

Recipe Corpus

KEY INGREDIENT:

ROOT VEGETABLES

Institute

50

New Yo

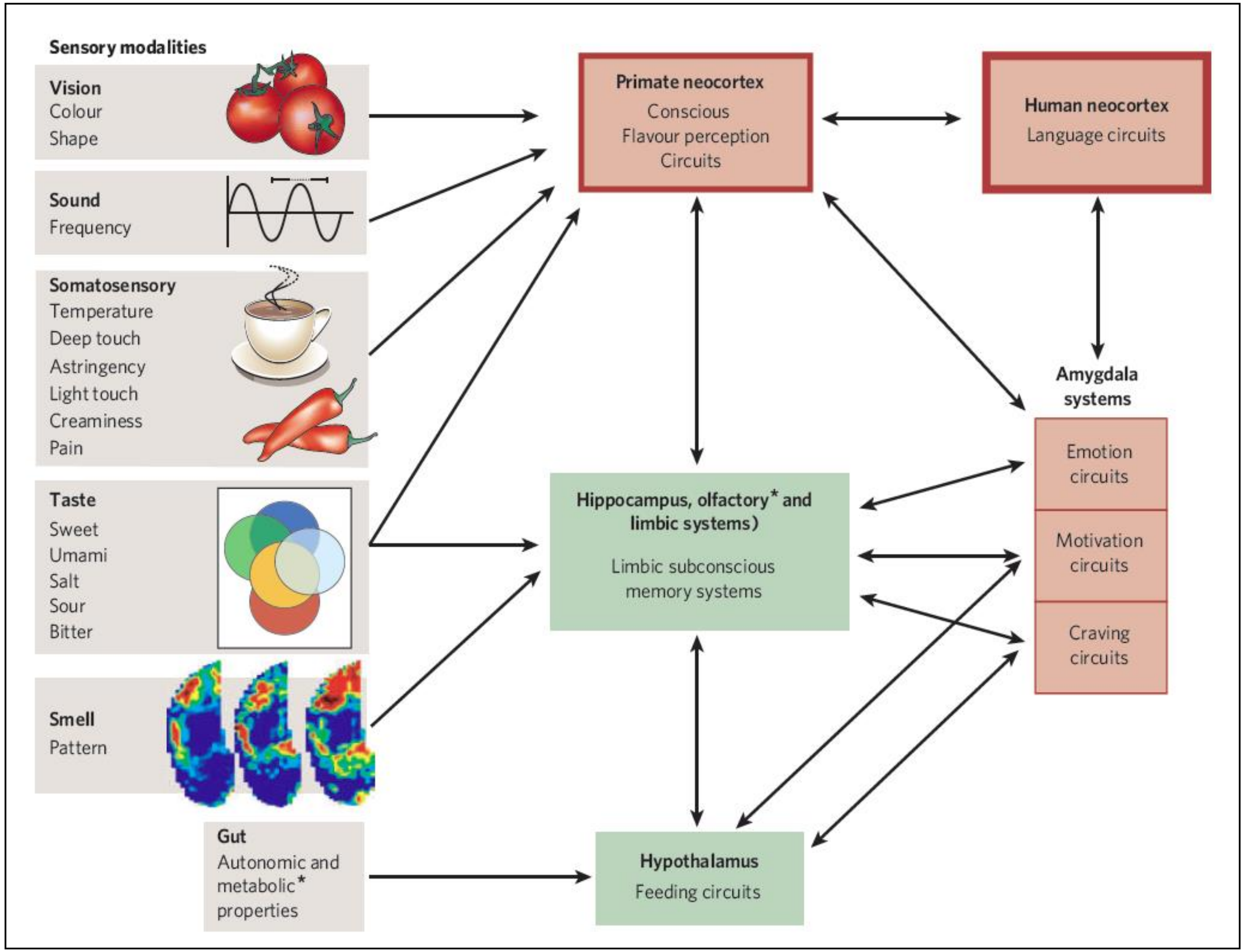
www

FRIED LOTUS ROOT CHIPS
Yield: Makes a lot

2 lotus roots, peeled
Vegetable oil to fry
Kosher salt to taste
Pinch of cayenne pepper, optional

Thinly slice the lotus root using a mandolin. (If not frying right away, hold the lotus root in water with some vinegar or lemon juice to prevent oxidation.) Heat 2 inches of oil in a heavy pot to 360° F. Pat the sliced lotus root dry with paper towel, and fry in batches until golden brown (they will continue to brown once removed, so cook just to golden). Transfer to a rack over a rimmed sheet pan, and sprinkle with salt (mix in a bit of cayenne pepper to the salt, if a spicier chip is desired).

Neurogastronomy

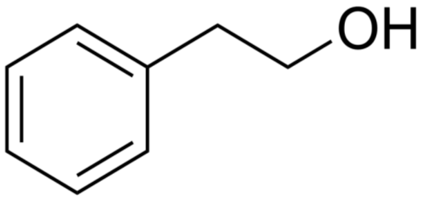


[Shepherd, 2006]

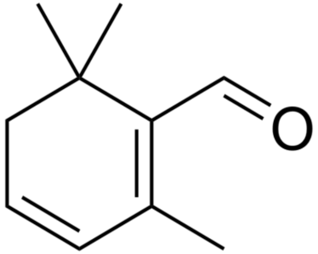
Food Chemistry

Saffron (*Crocus sativus L.*)

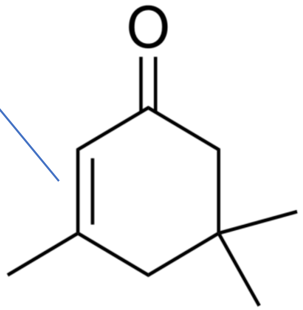
2-phenylethanol (=phenethyl alcohol)
safranal (=2,6,6-trimethyl-1,3-cyclohexadienecarbaldehyde)
3,5,5-trimethyl-2-cyclohexen-1-one (=isophorone)
hexadecanoic acid (=palmitic acid)
2,6,6-trimethyl-2-cyclohexene-1,4-dione
(Z,Z)-9,12-octadecadienoic acid (=linoleic acid)
(Z,Z,Z)-9,12,15-octadecatrienoic acid (=linolenic acid)
naphthalene
2,4,6-trimethylbenzaldehyde (=mesitylaldehyde)
2,6,6-trimethyl-1,4-cyclohexadienecarbaldehyde
6,6-dimethyl-2-methylene-3-cyclohexenecarbaldehyde
4-hydroxy-2,6,6-trimethyl-1-cyclohexenecarbaldehyde (=4-hydroxysafranal)
3,5,5-trimethyl-3-cyclohexen-1-one
3,3,4,5-tetramethylcyclohexanone
3,5,5-trimethyl-4-methylene-2-cyclohexen-1-one
4-hydroxy-3,5,5-trimethyl-2-cyclohexen-1-one
2,3-epoxy-4-(hydroxymethylene)-3,5,5-trimethylcyclohexanone
5,5-dimethyl-2-cyclohexene-1,4-dione
2,2,6-trimethylcyclohexane-1,4-dione (=3,5,5-trimethyl-cyclohexane-1,4-dione)
2-hydroxy-3,5,5-trimethyl-2-cyclohexene-1,4-dione
2-hydroxy-4,4,6-trimethyl-2,5-cyclohexadien-1-one
2,6,6-trimethyl-3-oxo-1,4-cyclohexadienecarbaldehyde
4-hydroxy-2,6,6-trimethyl-3-oxo-1,4-cyclohexadienecarbaldehyde
4-hydroxy-2,6,6-trimethyl-3-oxo-1-cyclohexenecarbaldehyde
3-hydroxy-2,6,6-trimethyl-4-oxo-2-cyclohexenecarbaldehyde
4-(2,2,6-trimethyl-1-cyclohexyl)-3-buten-2-one
4-(2,6,6-trimethyl-1-cyclohexen-1-yl)-3-buten-2-one (=β-ionone)
verbenone (=2-pinen-4-one)
octadecanoic acid (=stearic acid)
(Z)-9-octadecenoic acid (=oleic acid)
2(5H)-furanone (=crotonolactone, 2-buten-4-olide, 4-hydroxy-2-butenoic acid lactone)



phenethyl alcohol

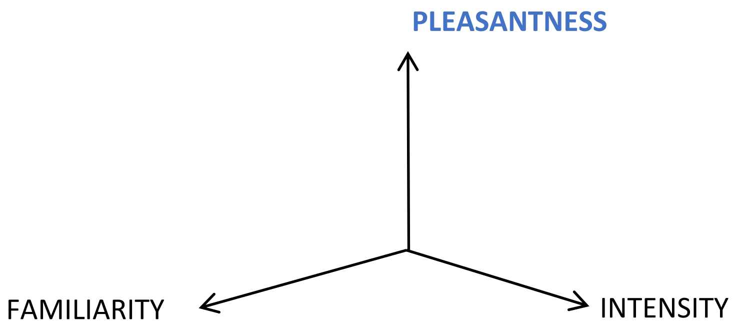


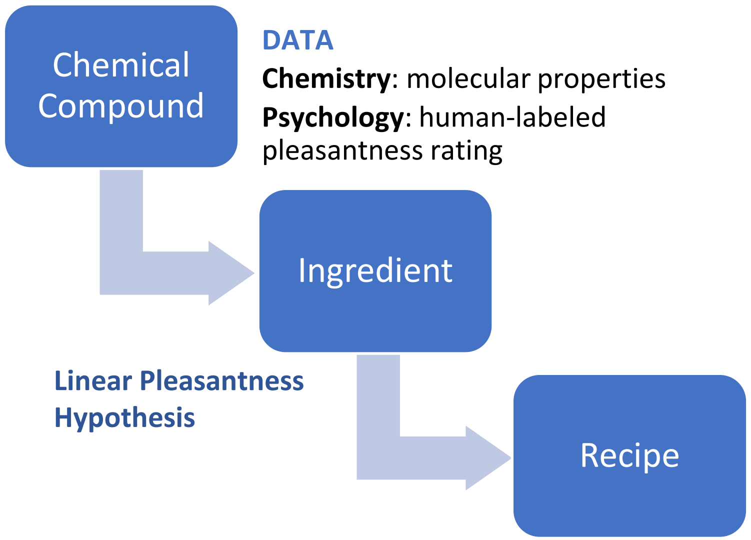
safranal

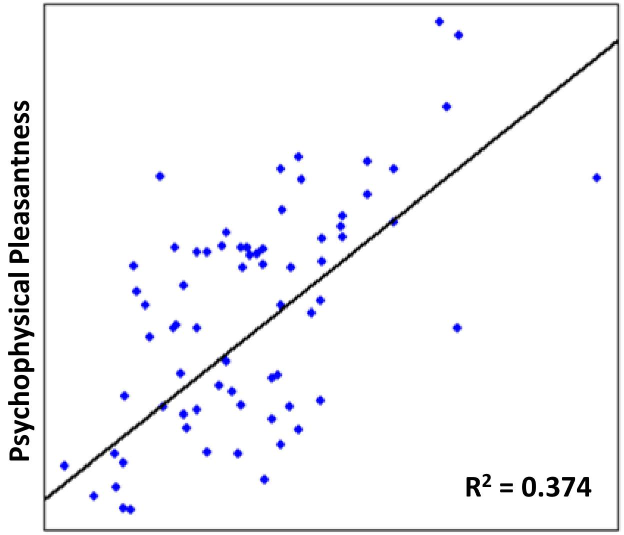


isophorone

Hedonic Psychophysics



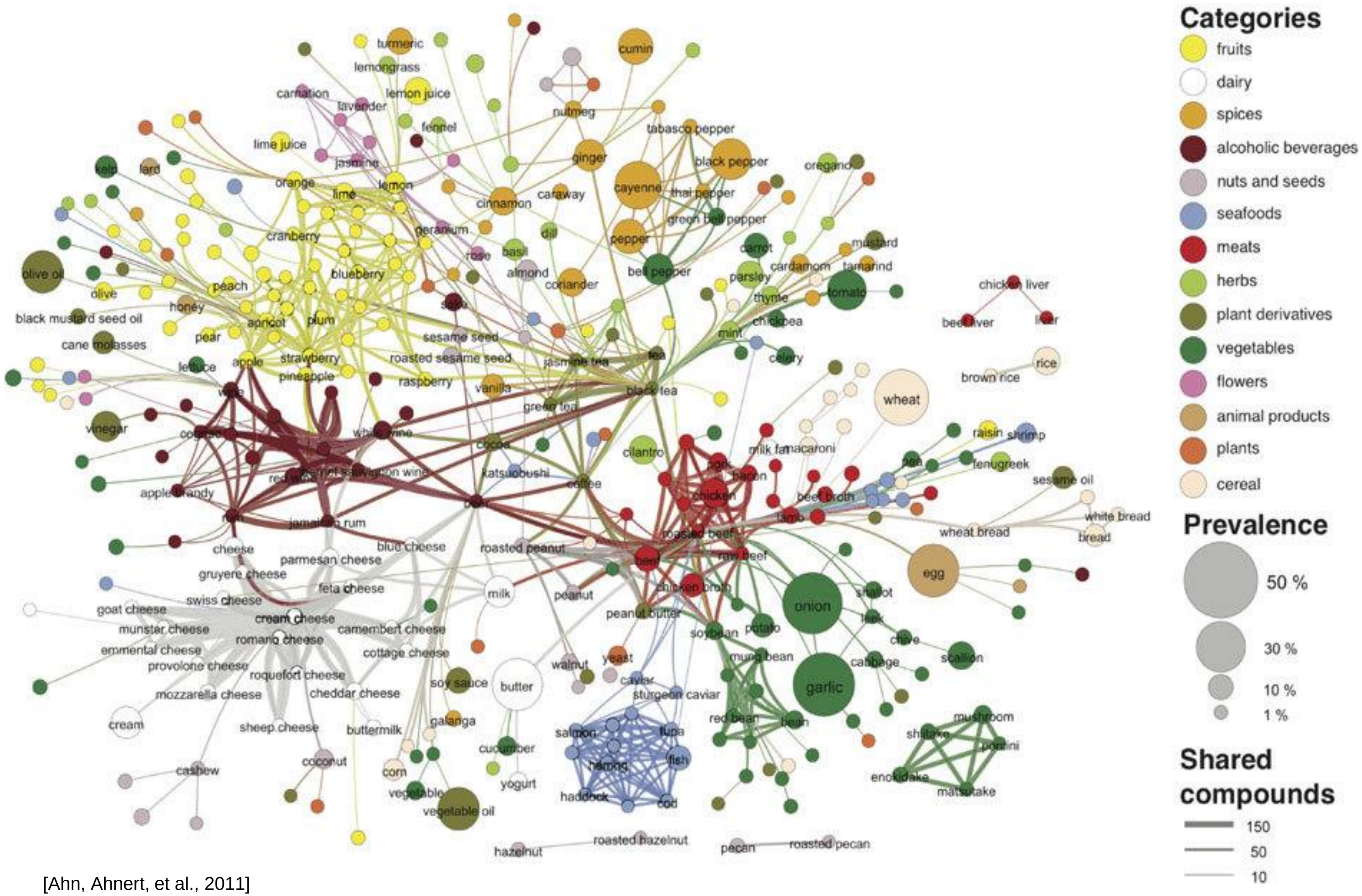




Chemistry [TPSA, heavy atom count, complexity, rotatable bond count, hydrogen bond acceptor count]

Black Tea
Bantu Beer
Beer
Strawberry
White Wine
Cooked Apple

Flavor Networks



[Ahn, Ahnert, et al., 2011]

Bayesian Surprise and Attention

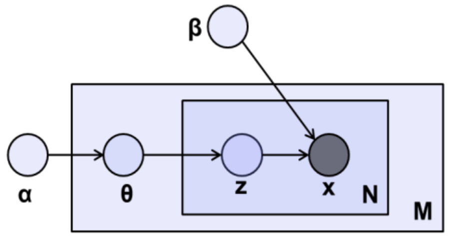
$$S(R,\mathcal{B}) = D\left(P_{B|R}||P_B\right) = \int_{\mathcal{B}} P_{B|R} \log \frac{P_{B|R}}{P_B} dB$$

newly created recipe

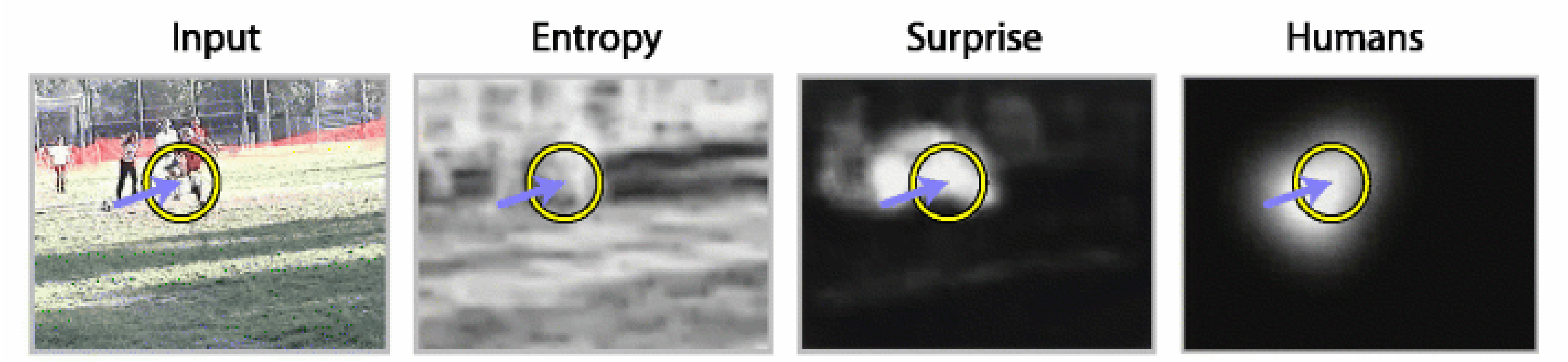
personalized repository of prior food experience

posterior beliefs

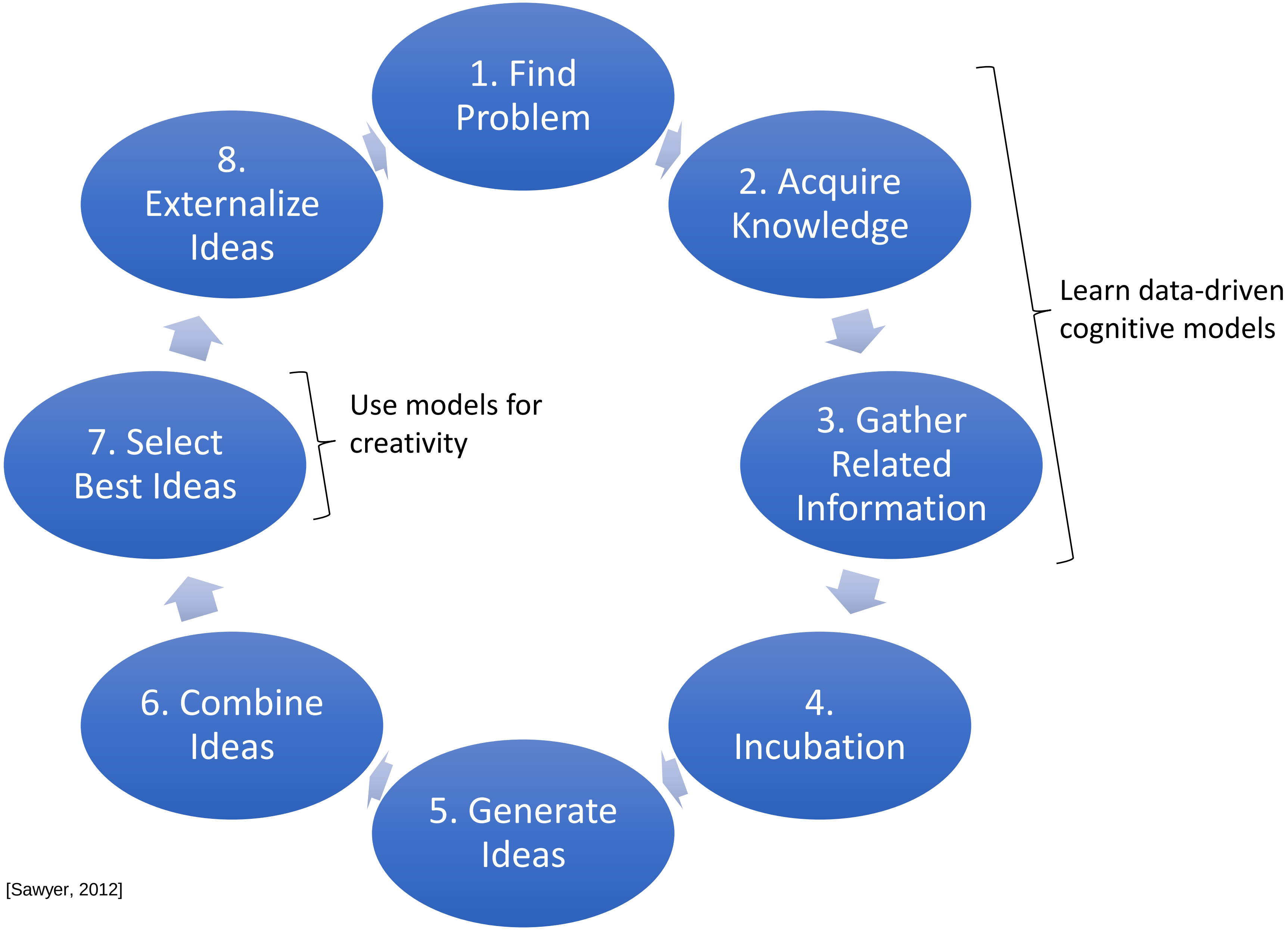
prior beliefs



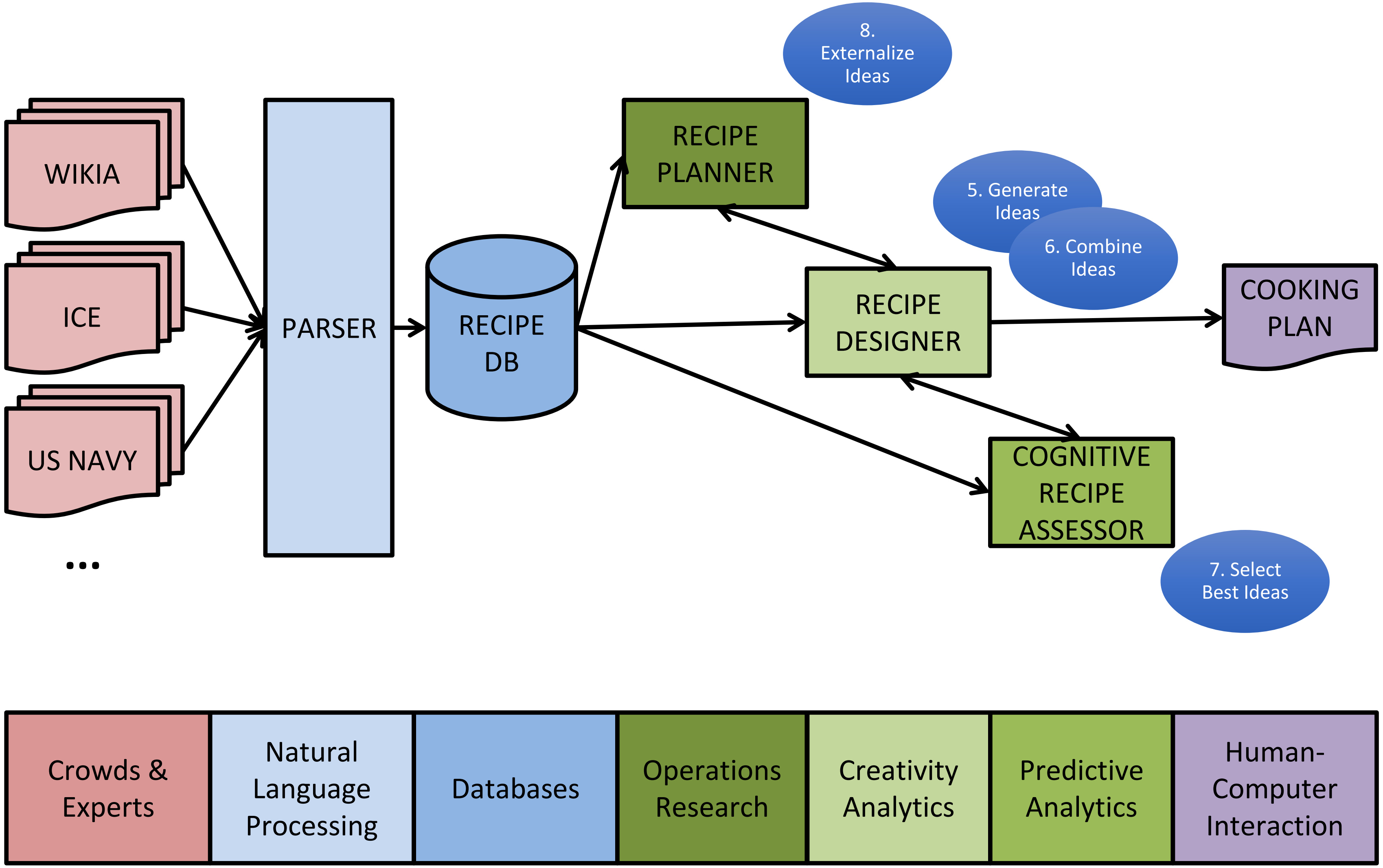
Latent Dirichlet Allocation (LDA) Model



[Itti and Baldi, 2006]


1. Find Problem
2. Acquire Knowledge
3. Gather Related Information
4. Incubation
5. Generate Ideas
6. Combine Ideas
7. Select Best Ideas
8. Externalize Ideas
Learn data-driven cognitive models
Use models for creativity

[Sawyer, 2012]



8. Externalize Ideas
WIKIA
ICE
US NAVY
...
PARSER
RECIPE DB
RECIPE PLANNER
5. Generate Ideas
6. Combine Ideas
RECIPE DESIGNER
COOKING PLAN
COGNITIVE RECIPE ASSESSOR
7. Select Best Ideas
Crowds & Experts
Natural Language Processing
Databases
Operations Research
Creativity Analytics
Predictive Analytics
Human-Computer Interaction

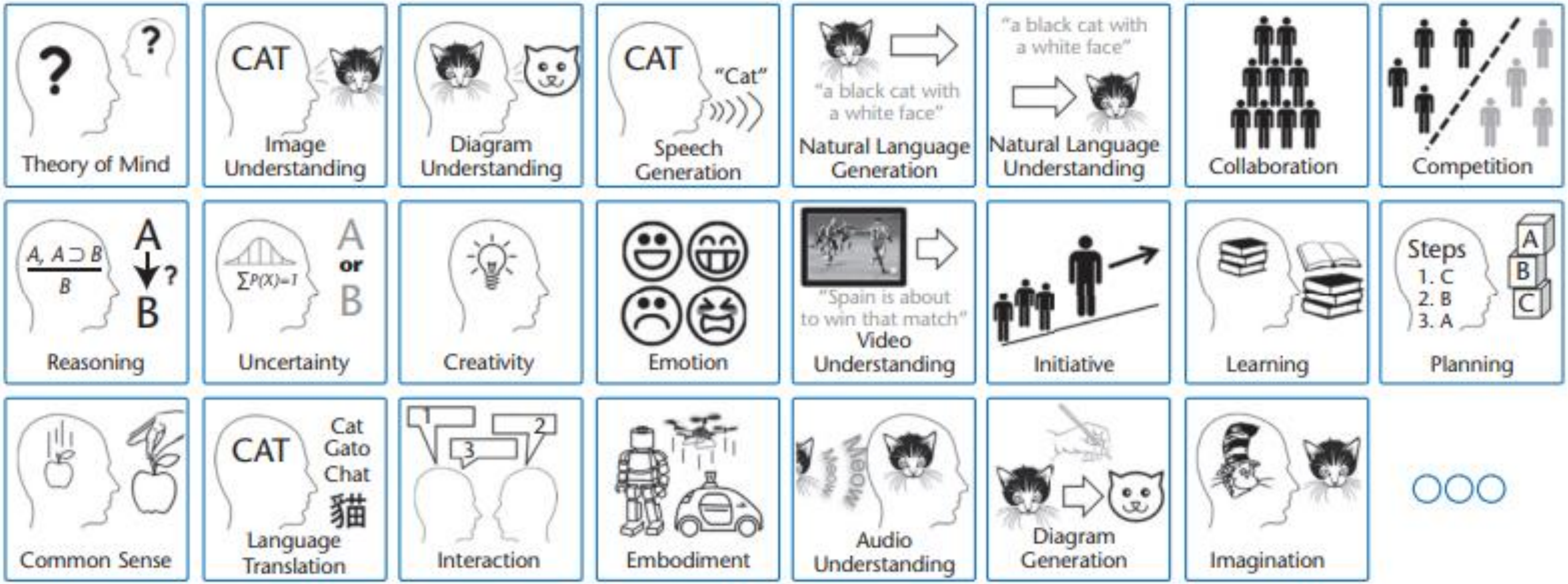
https://spectrum.ieee.org/ai-design

- Chef Watson fairly specific in terms of what representations are learned and what downstream purposes they can be used for
  - Olfactory pleasantness model can be used for perfume, indoor air quality, etc.
- Surprise model can be used in a fairly general-purpose way across modalities and domains

- Can learned representations be “general purpose technologies” for numerous tasks, like the steam engine or electricity?

I-athlon: Toward a Multidimensional Turing Test

Sam S. Adams, Guruduth Banavar, Murray Campbell



The Natural Language Decathlon: Multitask Learning as Question Answering

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, Richard Socher
Salesforce Research

Task	Dataset	# Train	# Dev	# Test	Metric
Question Answering	SQuAD	87599	10570	9616	nF1
Machine Translation	IWSLT	196884	993	1305	BLEU
Summarization	CNN/DM	287227	13368	11490	ROUGE
Natural Language Inference	MNLI	392702	20000	20000	EM
Sentiment Analysis	SST	6920	872	1821	EM
Semantic Role Labeling	QA-SRL	6414	2183	2201	nF1
Zero-Shot Relation Extraction	QA-ZRE	840000	600	12000	cF1
Goal-Oriented Dialogue	WOZ	2536	830	1646	dsEM
Semantic Parsing	WikiSQL	56355	8421	15878	lfEM
Pronoun Resolution	MWSC	80	82	100	EM

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*

Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry

Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan

Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter

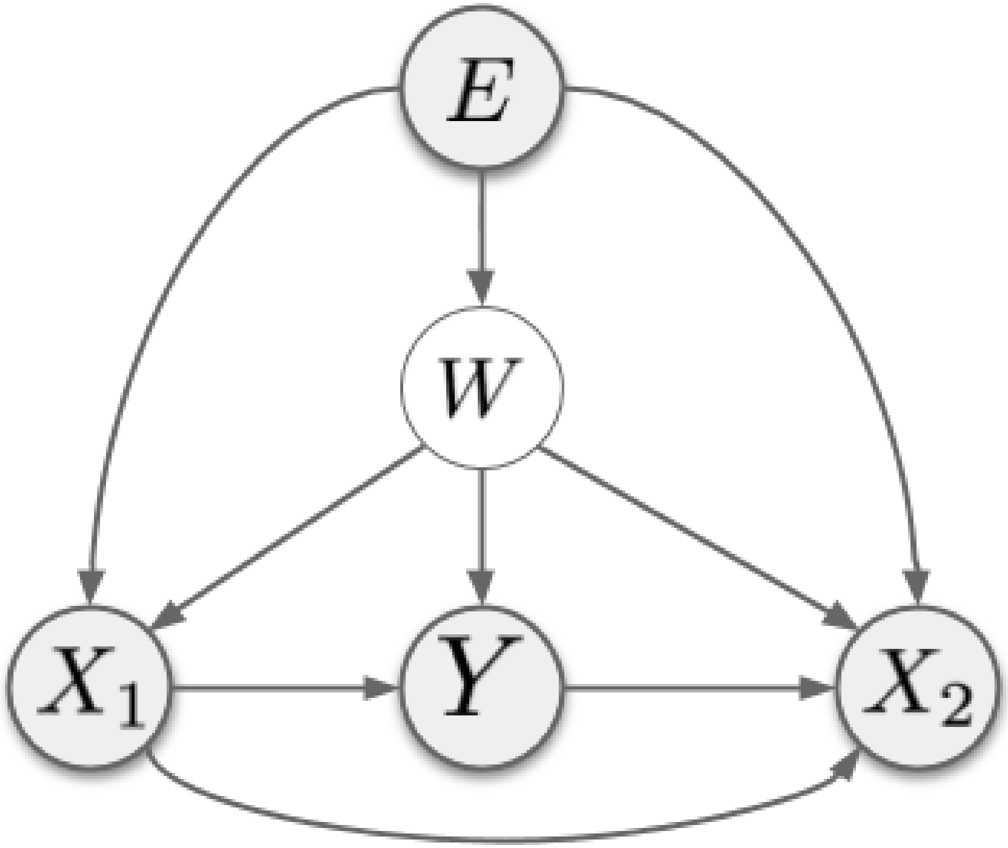
Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray

Benjamin Chess Jack Clark Christopher Berner

Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI

Environments



Tasks

entropy

Article

Information-Theoretic Generalization Bounds for Meta-Learning and Applications

Sharu Theresa Jose * and Osvaldo Simeone

Are there “invariant task representations”?

Meta-learning problem

- As formalized by the “no free lunch theorem”, any effective learning procedure must be based on prior assumptions on the task of interest
- These include the selection of a model class and of the hyperparameters of a learning algorithm, such as weight initialization and learning rate
- In conventional single-task learning, these assumptions, collectively known as inductive bias, are fixed a priori relying on domain knowledge or validation
- Fixing a suitable inductive bias can significantly reduce the sample complexity of the learning process, and is thus crucial to any learning procedure
- The goal of meta-learning is to automatically infer the inductive bias, thereby learning to learn from past experiences via the observation of a number of related tasks, so as to speed up learning a new and unseen task

In this work, we consider the meta-learning problem of inferring the hyperparameters of a learning algorithm. The learning algorithm (henceforth, called base-learning algorithm or base-learner) is defined as a stochastic mapping $P_{W|Z^m,u}$ from the input training set $Z^m = (Z_1, \ldots, Z_m)$ of m samples to a model parameter $W \in \mathcal{W}$ for a fixed hyperparameter vector u. The meta-learning algorithm (or meta-learner) infers the hyperparameter vector u, which defines the inductive bias, by observing a finite number of related tasks.

For example, consider the well-studied algorithm of *biased regularization* for supervised learning [9,10]. Let us denote each data point $Z = (X, Y)$ as a tuple of input features $X \in \mathbb{R}^d$ and label $Y \in \mathbb{R}$. The loss function $l : \mathcal{W} \times \mathcal{Z} \to \mathbb{R}$ is given as the quadratic measure $l(w, z) = (\langle w, x \rangle - y)^2$ that quantifies the loss accrued by the inferred model parameter w on a data sample z. Corresponding to each per-task data set Z^m, the biased regularization algorithm $P_{W|Z^m,u}$ is a Kronecker delta function centered at the minimizer of the following optimization problem

$$\frac{1}{m}\sum_{j=1}^{m} l(w, Z_j) + \frac{\lambda}{2}||w - u||^2, \tag{1}$$

which corresponds to an empirical risk minimization problem with a biased regularizer. Here, $\lambda > 0$ is a regularization constant that weighs the deviation of the model parameter w from a bias vector u. The bias vector u can be then thought of as a common "mean" among related tasks. In the context of meta-learning, the objective then is to infer the bias vector u by observing data sets from a number of similar related tasks. Different meta-learning algorithms have been developed for this problem [11,12].

In the meta-learning problem under study, we follow the standard setting of Baxter [13] and assume that the learning tasks belong to a *task environment*, which is defined by a probability distribution P_T on the space of learning tasks $\mathcal{T}$, and per-task data distributions $\{P_{Z|T=\tau}\}_{\tau\in\mathcal{T}}$. The data set Z^m for a task τ is then generated i.i.d. according to the distribution $P_{Z|T=\tau}$. The meta-learner observes the performance of the base-learner on the *meta-training data* from a finite number of *meta-training tasks*, which are sampled independently from the task environment, and infers the hyperparameter U such that it can learn a new task, drawn from the same task environment, from fewer data samples.

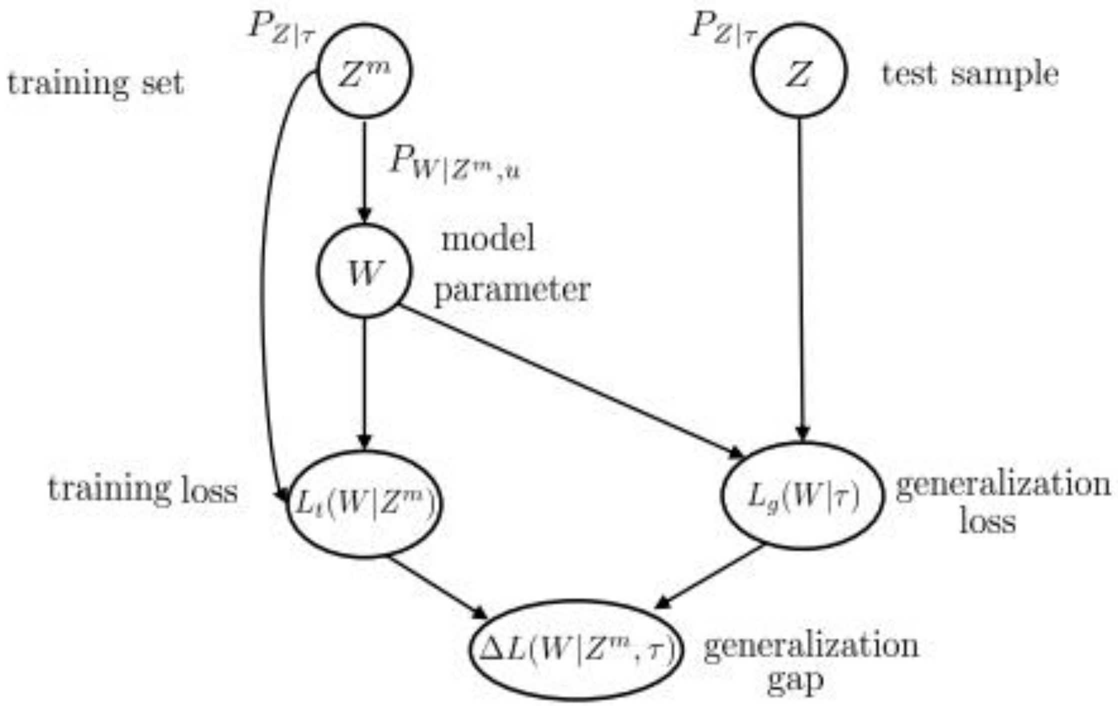
The quality of the inferred hyperparameter U is measured by the *meta-generalization loss*, $\mathcal{L}_g(U)$, which is the average loss incurred on the data set $Z^m \sim P_{Z^m|T}$ of a new, previously unseen task T sampled from the task distribution P_T. The notation will be formally introduced in Section 2.2. While the goal of meta-learning is to infer a hyperparameter U that minimizes the meta-generalization loss $\mathcal{L}_g(U)$, this is not computable, since the underlying task and data distributions are unknown. Instead, the meta-learner can evaluate an empirical estimate of the loss, $\mathcal{L}_t(U|Z^m_{1:N})$, using the meta-training set $Z^m_{1:N}$ of data from N tasks, which is referred to as *meta-training loss*. The difference between the meta-generalization loss and the meta-training loss is the *meta-generalization gap*,

$$\Delta\mathcal{L}(U|Z^m_{1:N}) = \mathcal{L}_g(U) - \mathcal{L}_t(U|Z^m_{1:N}), \tag{2}$$

and measures how well the inferred hyperparameter U generalizes to a new, previously unseen task. In particular, if the meta-generalization gap is small, on average or with high probability, then the performance of the meta-learner on the meta-training set can be taken as a reliable estimate of the meta-generalization loss.

In this paper, we study information-theoretic upper bounds on the *average meta-generalization gap* $\mathbb{E}_{P_{Z^m_{1:N}} P_{U|Z^m_{1:N}}}[\Delta\mathcal{L}(U|Z^m_{1:N})]$, where the average is with respect to the meta-training set $Z^m_{1:N}$ and the meta-learner defined by the stochastic kernel $P_{U|Z^m_{1:N}}$. Specifically, we extend the recent line of work initiated by Russo and Zhou [14], and Xu and Raginsky [15] which obtain mutual information (MI)-based bounds on the average generalization gap for conventional learning, to meta-learning. To the best of our knowledge, this is the first work that studies information-theoretic bounds for meta-learning.

Consider first the conventional problem of learning a task $\tau \in \mathcal{T}$. As illustrated in Figure 1



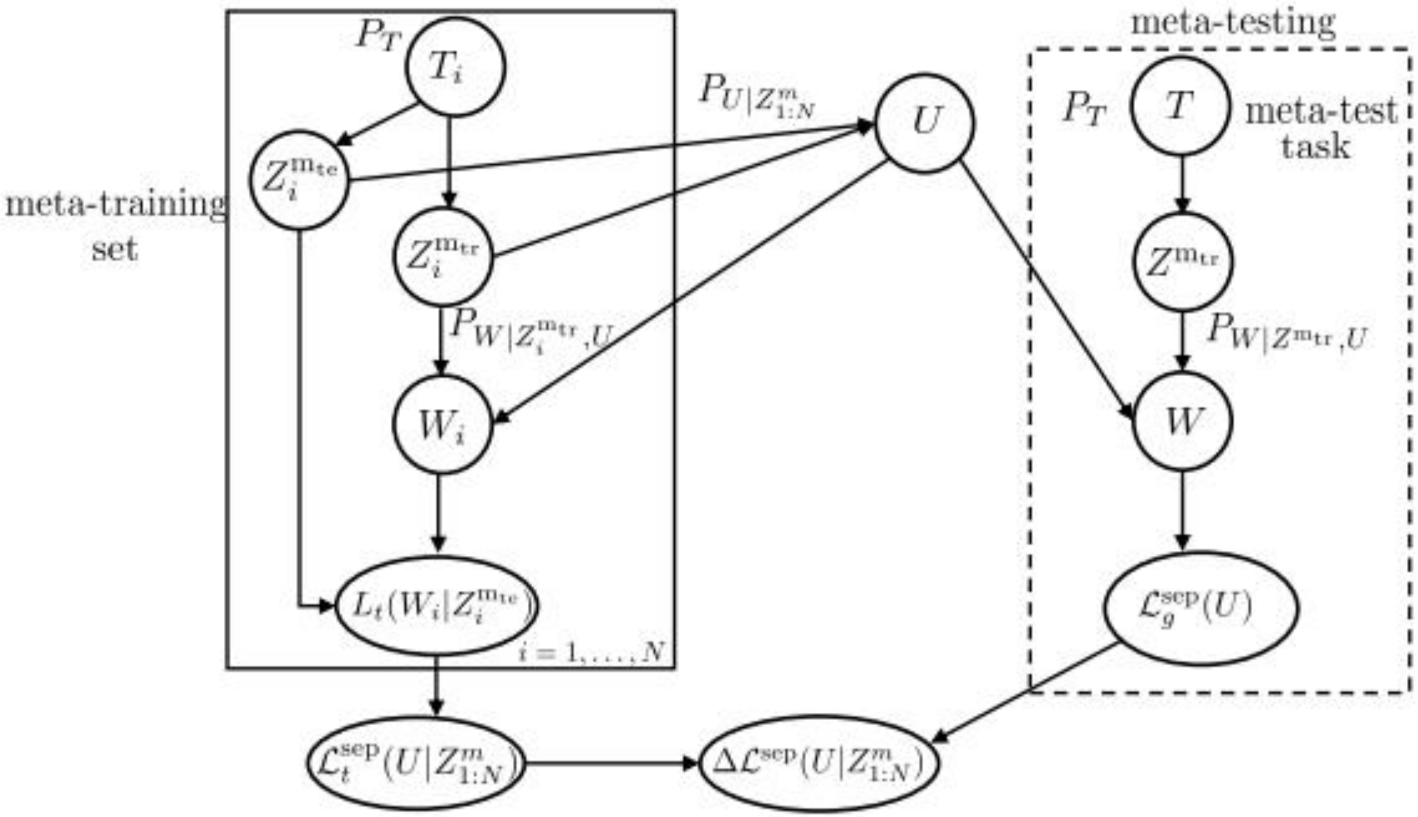


Figure 2. Directed graph representing the variables involved in the definition of meta-generalization gap (18) for separate within-task training and testing sets.

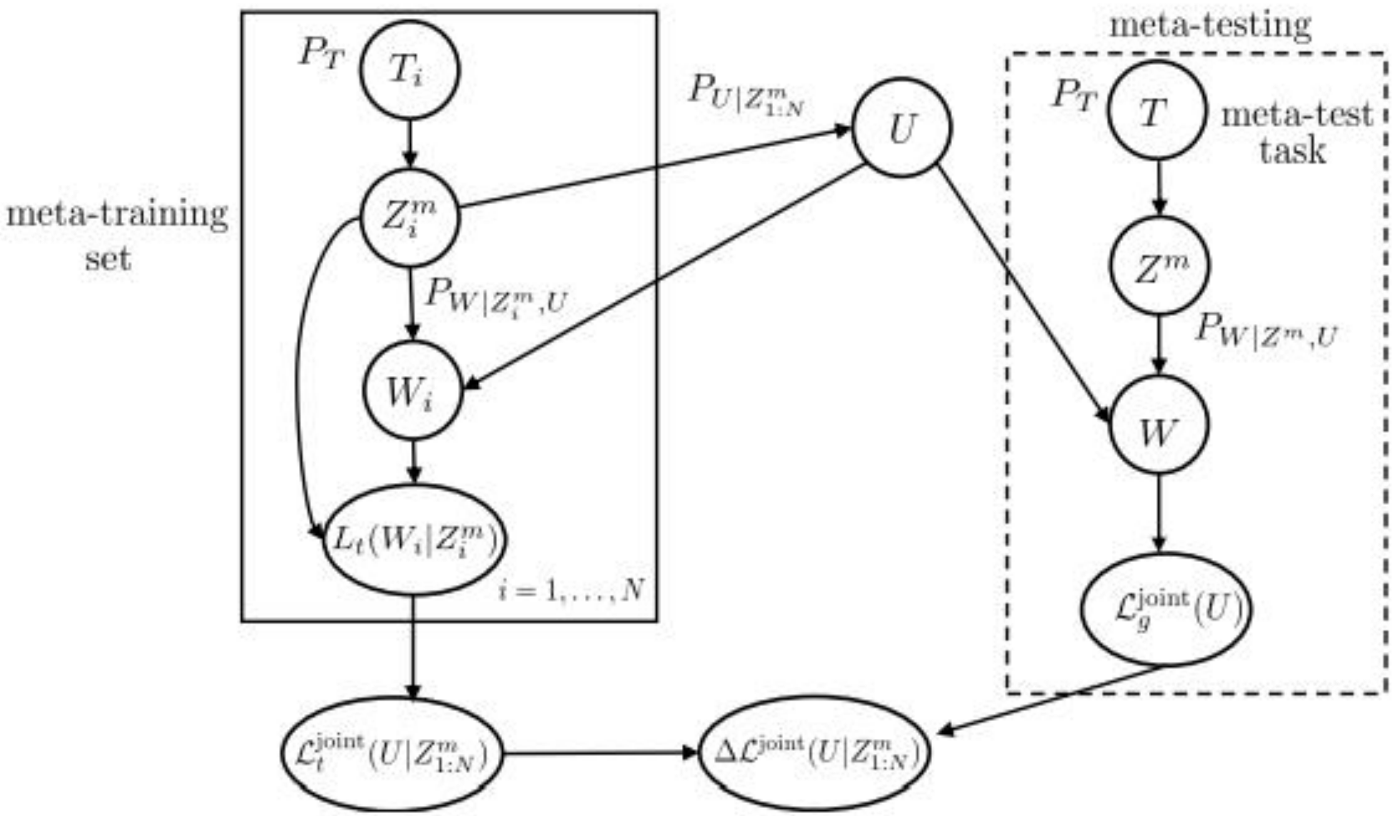


Figure 3. Directed graph representing the variables involved in the definition of meta-generalization gap (22) for joint within-task training and testing sets.

• In Theorem 1, we show that, for the case with *separate* within-task training and test sets, the average meta-generalization gap contains only the contribution of environment-level uncertainty. This is captured by a ratio of the mutual information (MI) between the output of the meta-learner U and the meta-training set $Z^m_{1:N}$, and the number of tasks N as

$$\left|\mathbb{E}_{P_{Z^m_{1:N}} P_{U|Z^m_{1:N}}}[\Delta\mathcal{L}^{\text{sep}}(U|Z^m_{1:N})]\right| \leq \sqrt{\frac{2\sigma^2}{N} I(U;Z^m_{1:N})}, \tag{3}$$

where σ^2 is the sub-Gaussianity variance factor of the meta-loss function. This is a direct parallel of the MI-based bounds for single-task learning [25].

• In Theorem 3, we then shown that, for the case with *joint* within-task training and test sets, the bound on the average meta-generalization gap also contains a contribution due to the within-task uncertainty via the ratio of the MI between the output of the base-learner and within task training data and the per-task data sample size m. Specifically, we have the following bound

$$\left|\mathbb{E}_{P_{Z^m_{1:N}} P_{U|Z^m_{1:N}}}[\Delta\mathcal{L}^{\text{joint}}(U|Z^m_{1:N})]\right| \leq \sqrt{\frac{2\sigma^2}{N} I(U; Z^m_{1:N})} + \mathbb{E}_{P_T}\left[\sqrt{\frac{2\delta_T^2}{m} I(W; Z^m|T=\tau)}\right], \tag{4}$$

where δ_T^2 is the sub-Gaussianity variance factor of the loss function $l(w,z)$ for task T.

How should one approach learning representations for several (and unknown) tasks