

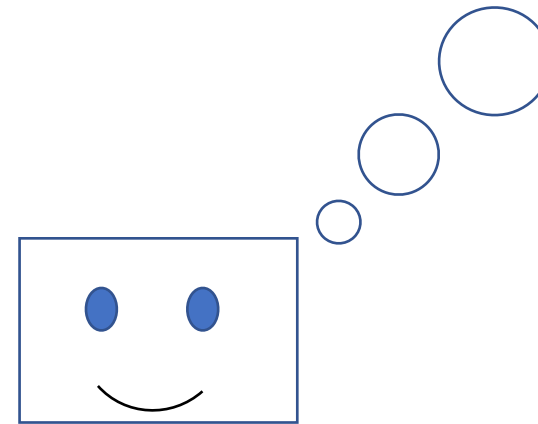
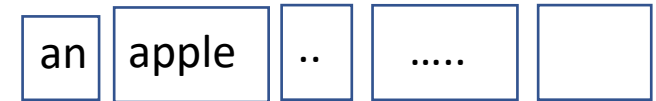
Advancing Prediction Games for Learning Networks of Hierarchical Patterns

(Can we bring some order (structure) back to ML?!)

Omid Madani

Cisco Secure Workload

www.omadani.net



↑
... anappleadaykeepsthe doctoraway ...

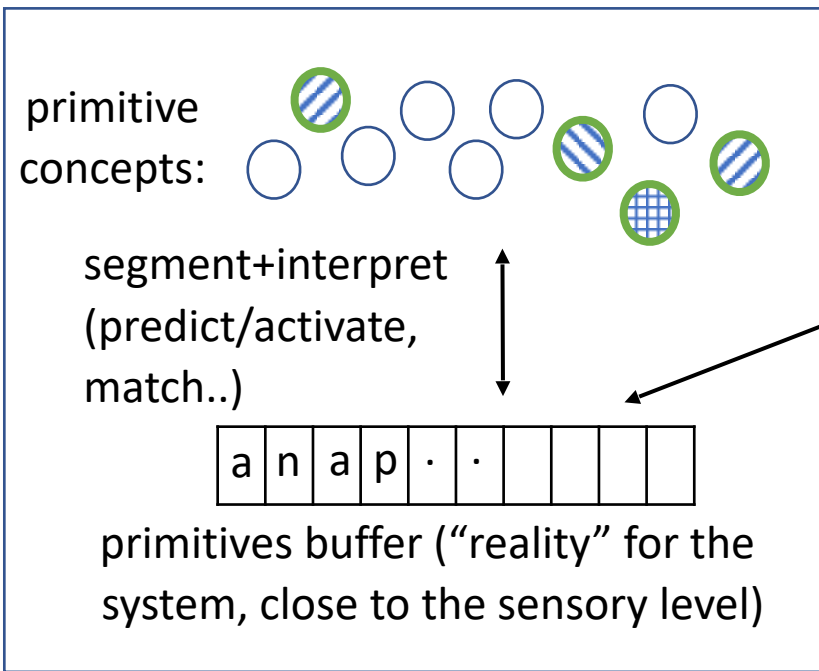
“Concepts are the glue that hold our mental world together... [but are] maddeningly complex.”

Gregory Murphy, “The Big Book of Concepts”

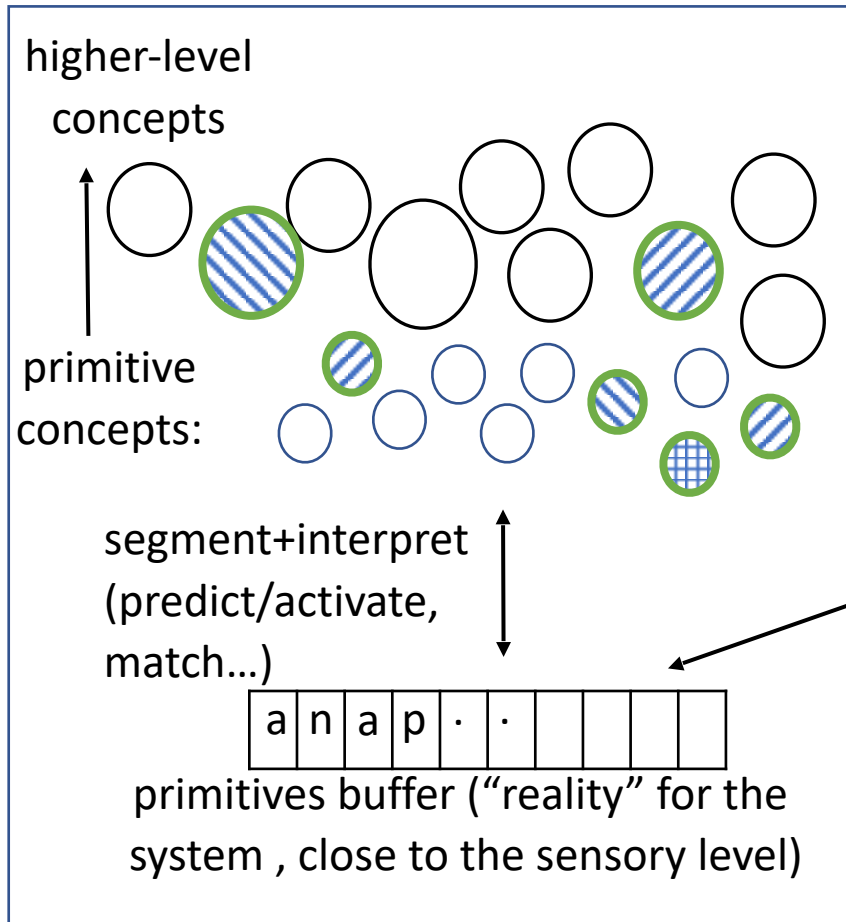
“.. to cut up each kind according to its species along its natural joints, ...”

Plato, “Phaedrus”


A Picture of Prediction Games (reality, concepts, and development)



After some learning



My Initial Rough Picture (in ~2000s)

- Repeatedly:
 - Get an episode (a scene, a line of text, ...)
 - Summon the relevant concepts (efficient recall)
 - Do (“conceptual”) analysis
 - Update (concepts) (.. and “put them back”!) 
- Possibly repeat (within a episode)

But what tasks/processes??!

(what does ‘relevant’ mean? Or ‘analysis’, ‘update’,.. or ‘concept’?)

Valiant, L. G. (1994). Circuits of the Mind.

Madani, O., Connor, M., and Greiner, W. (2009). Learning When Concepts Abound. *J. of Machine Learning Research* ([index learning](#))

Motivation/Philosophy/Approach

- How do humans acquire so many concepts (apparently)?
- How do we reach common sense?
 - Without explicit labels (largely unsupervised)..
 - Situated/immersed in a noisy complex world ..
 - What are the tasks, processes, etc. ? ... (how could ml help?!)
- Approach:
 - Unsupervised/immersed!
 - Build own concepts as prediction targets and as predictors (self-supervised!)
 - concept \approx spatiotemporal pattern, plus connections..
 - The world is (often) hierarchical!
 - Prediction is the driver: validate concepts by prediction/matching



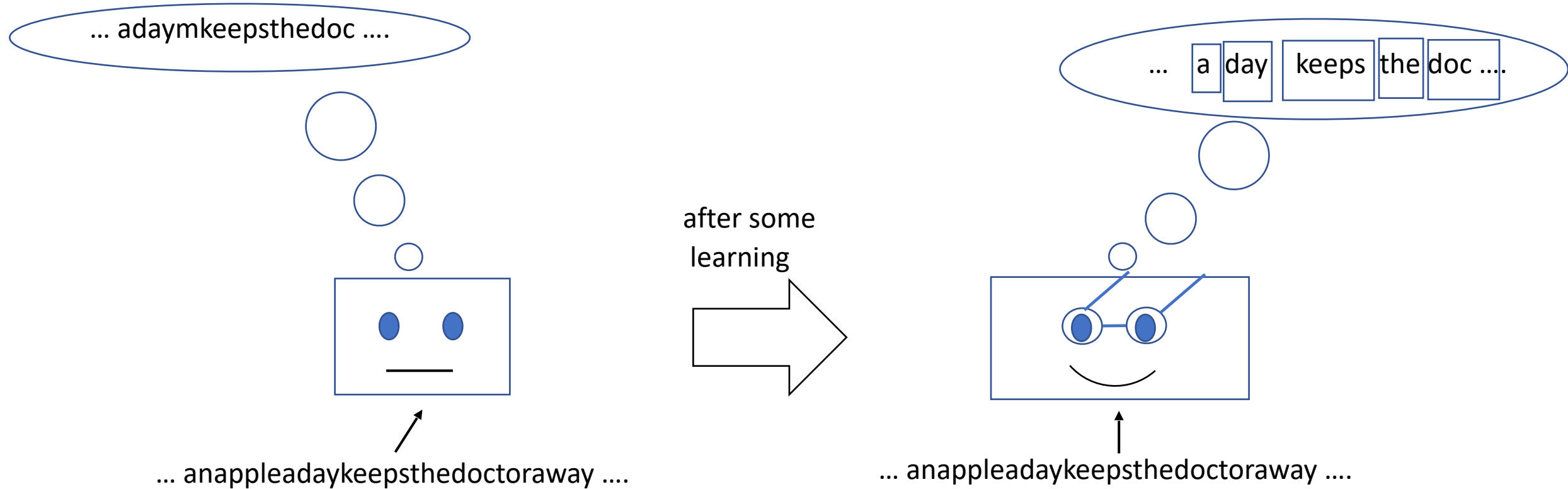
Madani, O. (2007). Prediction Games in Infinitely Rich Worlds. In *AAAI Fall Symposium*.

Kru'ger, N. et. al. (2013). Deep Hierarchies in the Primate Visual Cortex: What can we learn for computer vision? *IEEE PAMI*

Bubic', A. et. al. (2010). Prediction, cognition and the brain. *Frontiers in Human Neuroscience*

Hawkins, J. and Blakeslee, S. (2004). *On Intelligence: How a New Understanding of the Brain will lead to Truly Intelligent Machines*

A Simple Picture of Prediction Games (Development)



With learning:

- The vocabulary, or the **hierarchical network**, of (structured) concepts grows
- The system *interprets* the input in terms of higher level concepts (to make better sense of the world)..

Lifelong learning, unsupervised, immersed/situated in the real, richly structured, sensory world.

A system that makes its own concepts, to use 1) as *predictands* (prediction targets) and 2) as *predictors*, and 3) as *building blocks* for higher-level concepts.

The system is composed of multiple learning and inference processes, interacting..

Potential Benefits

(of Structure Learning)

- Prediction (but feed-forward neural nets do that!)
- More robustness
- Interpretability
 - *(no interpretability without machine interpretation?!)*
 - Will interpretability last? (as we attempt to extend concept structure?)
- Learning patterns over concepts' structures (a kind of meta learning)
- Faster generalization, fewer tail issues
- Symbols! Communication (sharing experience)
- (can call it: a discrete NN, or SLM with growing hierarchical vocab, ...)

A Fast (< 10 Minutes) Tour!

Overview of Computation

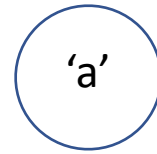
- Begin with **primitive concepts** (the alphabet) corresponding to, say, characters, no edges (*tabula rasa*)
- Repeat
 - Get next input, a line of text
 - **Segment + interpret** : which concepts are **active** (and where)?
 - Involves prediction/inference/search (beam search) ...
 - **Learn**: update active concepts' seen counts, prediction weights, ...)
- Periodically do the **offline phase** tasks: build new concepts, other (possible 'global') operations, house keeping, etc...

Outcome:

- Learned data structure: hierarchical network of concepts (associations and part edges)
- Function: interpretation, ie breaks an episode (line) into active concepts

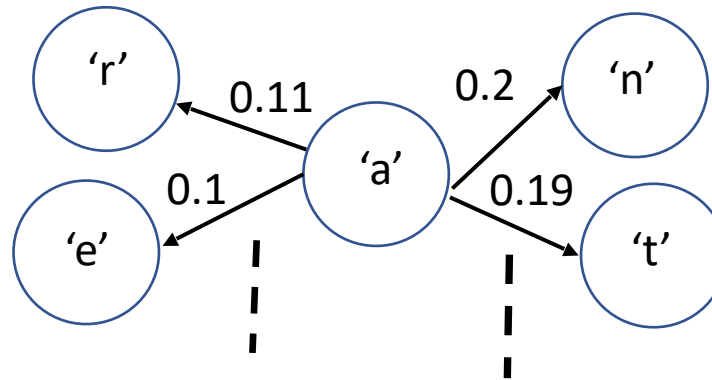
Initially... Tabula Rasa

- Begin with a set of *primitive concepts* (*alphabet, or initial vocabulary*), corresponding to, say, single characters,



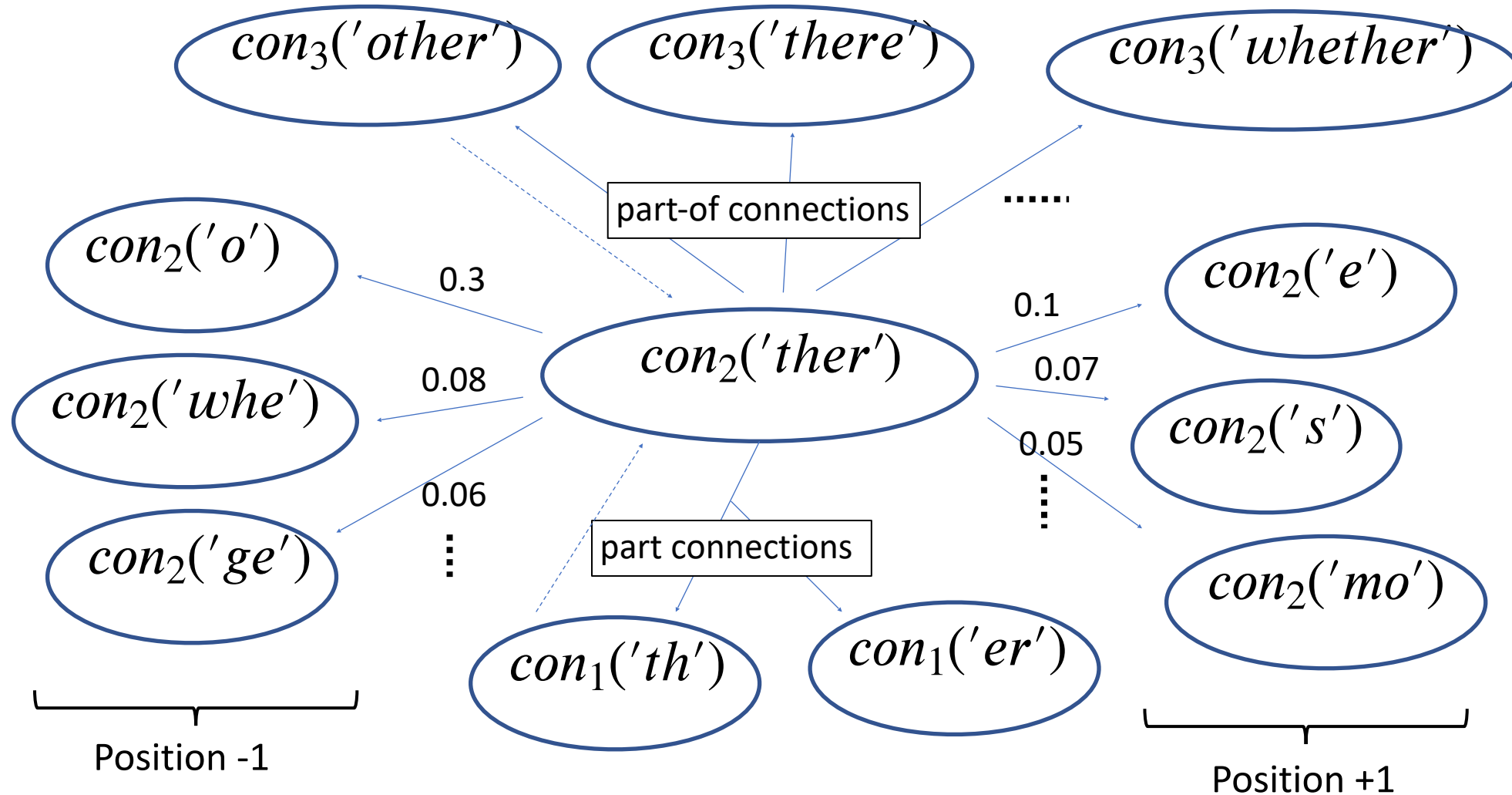
(an example concept, a single node, initially no edges..)

- After some time:



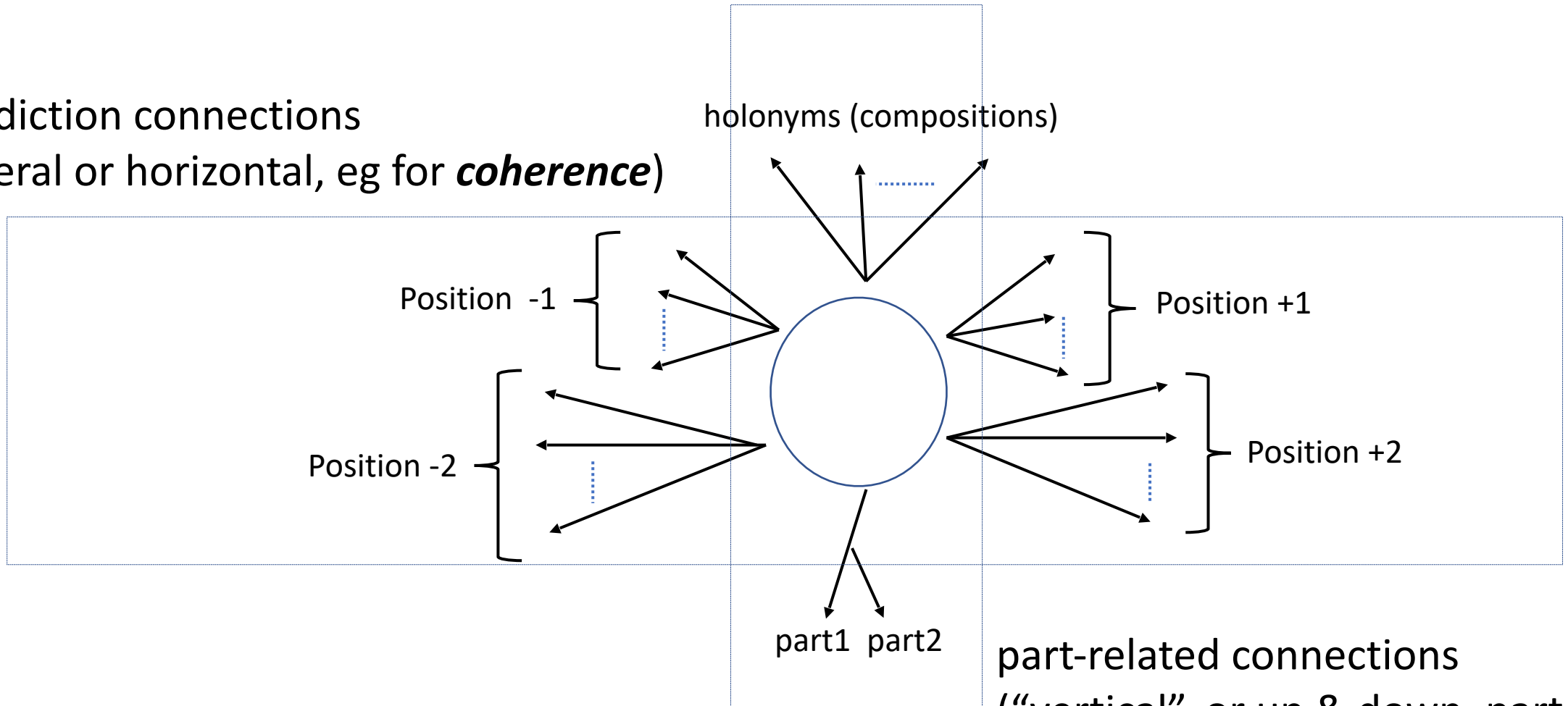
- After more time,
higher level concepts too ..

An Example Concept's Structure & Connections (After Some Learning)



Anatomy of a Concept's Connections

Prediction connections
(lateral or horizontal, eg for *coherence*)



part-related connections
("vertical", or up & down, part and
part-of, used for *invoking/matching*)

An Example Interpretation

(after some training)

(up to 8-grams)

Level 3: reg arding the conser vation and mana gement ofthese magni fice nt



(up to 4-grams)

Level 2: re g ar ding the con ser va tion and ma na ge ment oft hese mag ni fice nt



(up to bigrams)

Level 1: re g ar di ng t he c on se r va ti on a nd ma n a ge me nt o ft he se ma g n i fi ce nt



(unigrams)

What system sees, level 0: "regardingtheconservationandmanagementofthesemagnificent"

Original Input line: "regarding the conservation and management of these magnificent"

Hierarchy and Context

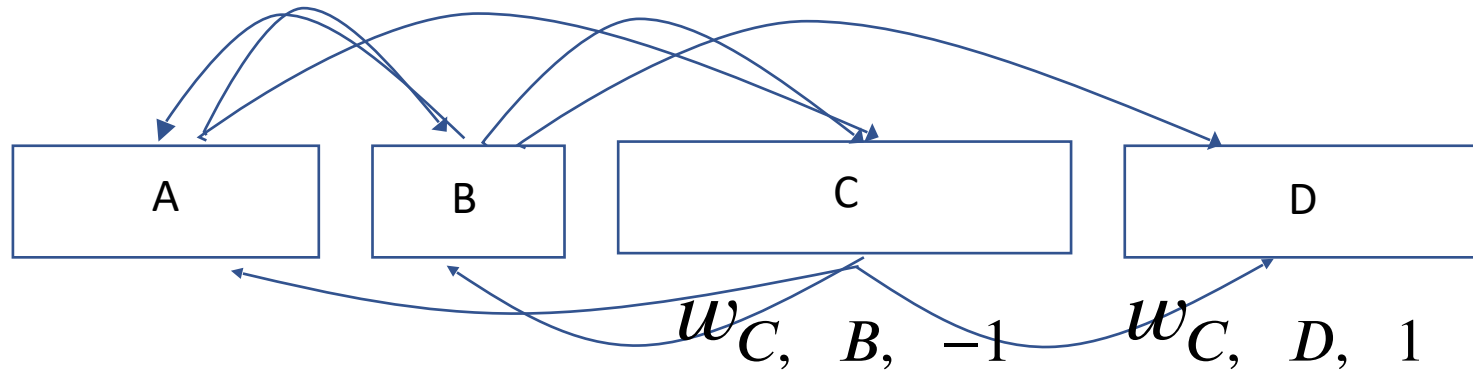
- The character 'a' appearing in the input stream will always activate the primitive concept corresponding to 'a'
- But 'a' can be high level "a" too (an indefinite article in English)
 - Depends on context
- The hope is that a concept from a high enough level "a" will only or mostly activate when the 'a' in the input corresponds to the (isolated) article "a".
 - Example: "I read a book!"

(a high level concept "a" should be activated by the 2nd occurrence!)

Two different meanings of 'a'

Updating, Given an Interpretation

- Lateral association weights updated via *sparse EMA* (Exponential Moving Average)
 - Active concepts update for each position around them
 - Active concepts also update their seen counts, probabilities received,...



B strengthens its weight to A at position -1
B strengthens its weight to C at position +1
B strengthens its weight to D at position +2
A strengthens its weight to B at position +3
A strengthens its weight to C at position +2
.....

} All via EMA

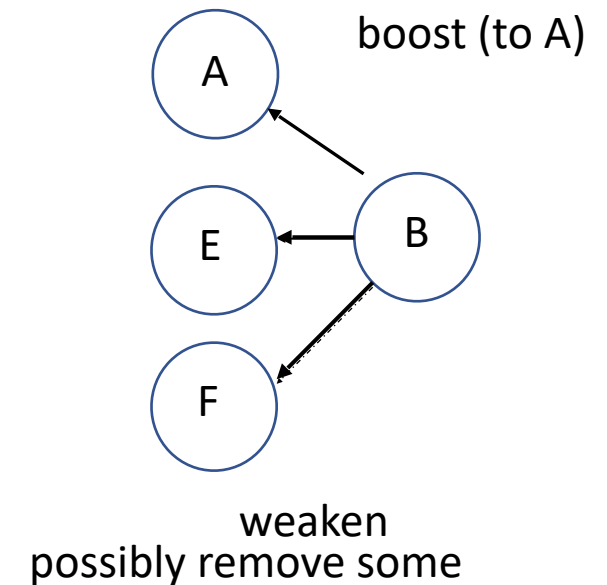
(The prediction window size is 2)

Sparse EMA, for Prediction-Weight Updates

- A simple (elegant!) *moving average* (for non-stationarity) ...
 - Versatile, we use it in several ways ...
- Concepts use EMA for sparse updates of own prediction weights
- EMA (for edges):
 - Weaken all (existing) edges by $1 - \beta$, or $w \leftarrow (1 - \beta)w$
 - Boost connection to observed concept by β , $w \leftarrow w + \beta$
 - Insert edge, if not there (absent edges have 0 weight)
 - Once in a while, prune weak edges
- Do this for all positions: $\pm 1, \pm 2, \dots, \pm WindowSize$

O. Madani and J. Huang (2008). On updates that constrain the number of connections of features during learning. In *ACM KDD*

O. Madani, H. Bui, and E. Yeh (2009). Efficient Online Learning and Prediction of Users' Desktop Behavior. In *IJCAI*



Some EMA Properties

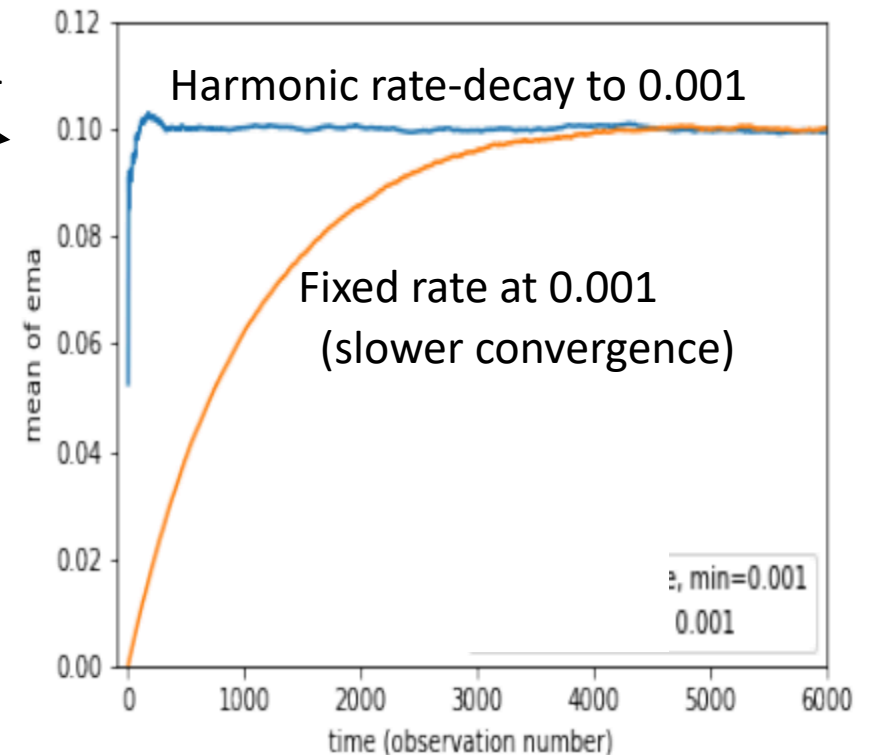
- Edge weight converges to $P(A|B)$, when $\beta \in (0,1)$ is sufficiently small
 - The edge weights form a “sub-distribution”
- Use higher rates for new concepts (*e.g.*, do ***harmonic rate decay***).

Such as: $\beta \leftarrow \max((\beta^{-1} + 1)^{-1}, 0.001)$
(why? Because faster convergence!)

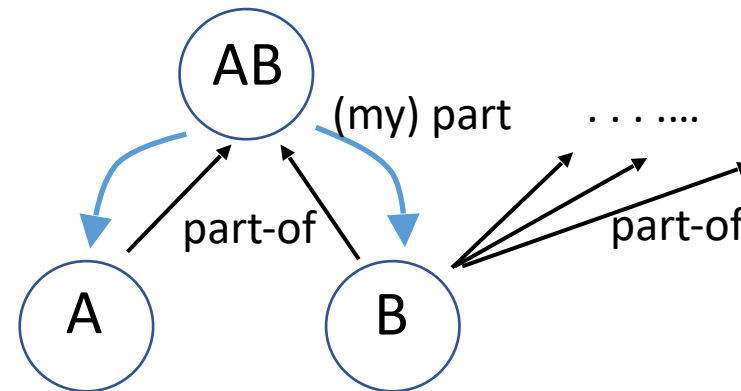
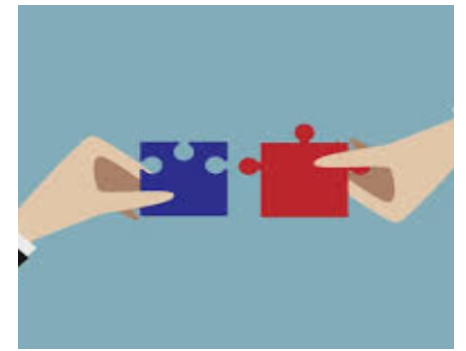
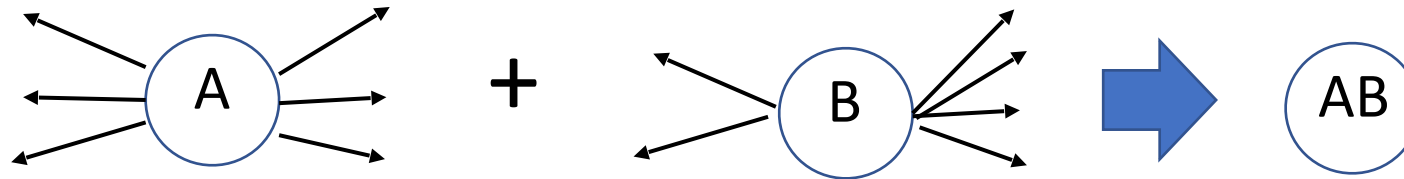
True probability is 0.1

Note: Each concept has its own rate!
Future: even more dynamic rates

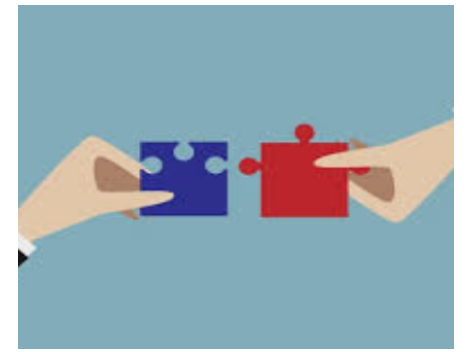
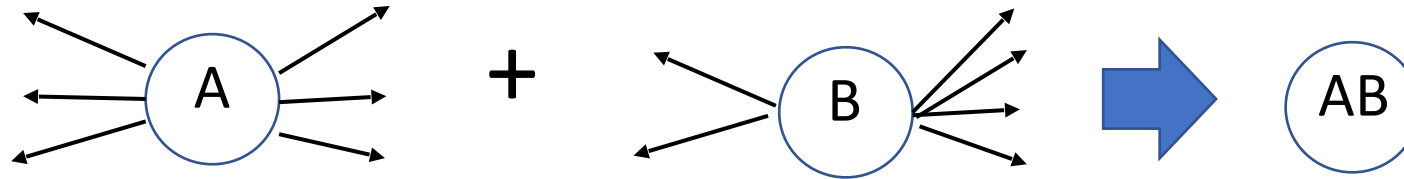
O. Madani (2021). Expedition: A System for the Unsupervised Learning of a Hierarchy of Concepts, ArXiv.



Composing (Offline)



Composing



- Used the binomial tail for $P(A|B) > P(A)$, ie sufficiently strong evidence that conditional is larger than prior..
- Notes:
 - Doesn't guarantee belonging to same concept.. But may suffice
 - Make bigrams only (local unary & binary operations only)
 - Need an ***Exploratory period*** for new concepts (use the new concept a bit) ..
- Whether to compose/update at all levels, or say top level only (pros/cons)?
 - We have explored both, but composing only at top appears simpler..



Segmenting/Interpreting
and how to promote use of larger concepts

CORE: Quantifying Value (of Prediction, ...)

1. Survival advantage (of predicting larger patterns):

An organism that can predict further into the future (or farther, in space) ***in one shot*** is better off (than the more myopic...)

2. (Observation) Larger patterns are better predictors:

'z' (in English text) has significant prediction power over a few characters next to it, but "zoo" has significant predictive power over several word locations around it.

(and the whole, often, is more than the sum of its parts)

- How to quantify the prediction value of a concept, in an episode, and on average??
And what about the quality of an interpretation?

CORE: How Much Better than a Baseline System?

Concept C

characters of C

$$\text{MatchReward}(\mathcal{C}) = -\log\left(\prod_{1 \leq i \leq k} \text{prior}(c_i)\right) = -\sum_{1 \leq i \leq k} \log(\text{prior}(c_i)) \quad (\mathcal{C} \text{ corresponds to } c_1 \cdots c_k).$$

(coherence) probability assigned by system to C

$$\text{CORE}(\mathcal{C}) = \log\left(\frac{\text{pred}(\mathcal{C})}{\prod_{1 \leq i \leq k} \text{prior}(c_i)}\right) = \underbrace{\log(\text{pred}(\mathcal{C}))}_{\text{coherence}} + \underbrace{\text{MatchReward}(\mathcal{C})}_{\text{match (to reality)}}$$

In general, probability assigned to c by some baseline/reference system (in our case, we are using the character level indep. assumption predictor).

CORE = COherence + REality



match to reality

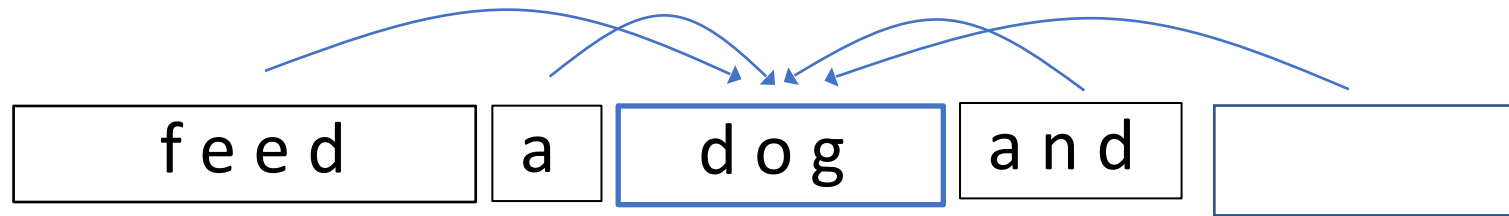
primitives level ("reality")



Combining: Fit to Context and Matching “Reality”

coherence (prediction strength)

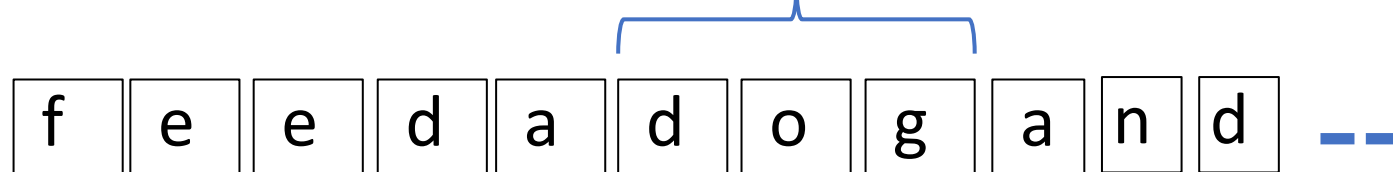
highest level:



1. Higher, better
2. Related to entropy..
3. A “positive” score (unlike perplexity/entropy)

match (to “reality” or ground-level)

primitives level:



Generalizing CORE to Generative Models

(concepts as generative models)

Generation probability of observed sequence T by concept C

$$\text{MatchReward}(\mathcal{C}, T) = \log\left(\frac{P_C(T)}{\prod_{1 \leq i \leq k} \text{prior}(t_i)}\right) \quad (\text{where string } T = t_1 \cdots t_k)$$

$$\text{CORE}(\mathcal{C}, T) = \underbrace{\log(\text{pred}(\mathcal{C}))}_{\text{coherence}} + \underbrace{\text{MatchReward}(\mathcal{C}, T)}_{\text{match (to reality)}}$$

A Successful Match is not Enough

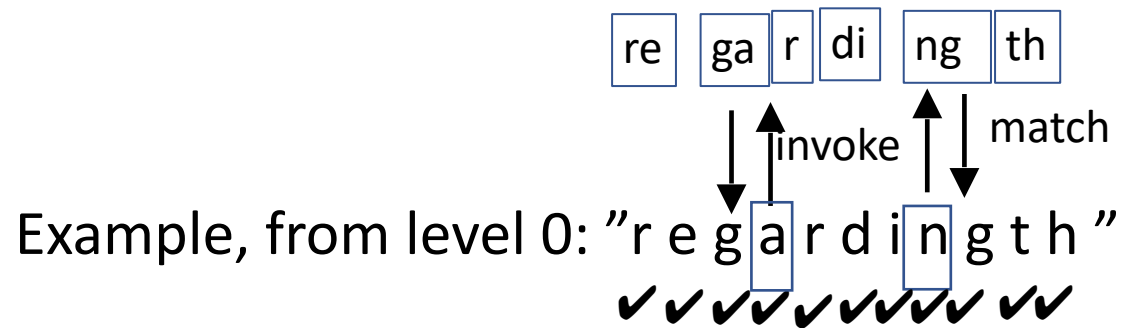
- Learning when to use a concept...
 - Matching is not enough (even if the concept is good/useful in general) (however, it could be a strong signal)
 - A chicken-and-egg problem!

Matching concept “en” is a mistake here

m o v e n o w

Interpretation Search

- Repeat
 - Repeatedly pick a random position/concept, invoke holonyms, match
 - Until no more options left (e.g. no matching holonyms)
- Pick best interpretation via average CORE



Summary Empirical Findings

- Trained on NSF abstracts (UCI dataset)
- N-grams learned mostly look good!
 - 1000s of concepts after hundreds of thousands of episodes
- Splits in segmentations correspond well with word boundaries
 - More training or inference time helps increase the good splits
- Log-loss on character prediction compared with classic n-grams for SLM, and transformer NNs, competitive on one type of task (1st-letter of word prediction), more work needed on others

Examples, Most Seen Concepts at Each Length (trained on 20k lines, a few passes)

1-gram	2-gram	3-gram	4-gram
e	ti	and	tion
t	he	pro	will
i	th	ate	ment
o	on	ati	fthe
s	st	ing	with

Character Prediction, Log-Loss

- The lower the better
- Trained on NSF abstracts
- Tested on next character prediction, in middle of the line
 - 1st letter of a word
 - Last letter of a word
 - Random ('rand')

tasks (on 200 lines) →	1st	last	rand	1st	last
methods ↓	NSF			NEWS	
Train on 500 lines (500 data)					
unigrams only	3.84	2.2	2.7	4.1	2.85
up to bigrams	3.82	1.7	2.5	4.3	2.75
NGR (all n-grams)	3.90	1.4	2.4	4.4	2.7
NN (Transformer)	3.76	2.1	2.6	4.0	2.6
Expedition, 2 rounds	3.79	2.4	2.7	4.1	2.9
Expedition, 3 rounds	3.89	2.6	2.8	4.2	3.0
Train on ≈21k lines (20k data)					
unigrams only	3.72	2.2	2.7	3.8	2.6
up to bigrams	3.49	1.6	2.4	3.9	2.45
NGR (all n-grams)	3.47	0.68	1.8	4.3	2.3
NN (Transformer)	3.22	0.76	1.6	4.1	2.1
Expedition, 2 rounds	3.43	2.1	2.5	3.9	2.7
Expedition, 3 rounds	3.52	2.1	2.6	3.95	2.7

Summary Features

- Open ended learning
 - Sparse network, growing without bound (with more training/experience)
- Concepts (nodes) as sophisticated book keepers: they keep much state, etc. (but efficient)
- Complex costly interpretation (code/engineering complexity, data structures, etc.)
- A systems approach: multiple processes interacting (even for 'low-level' pattern recognition tasks)

Is the cost/complexity worth **explicit structure learning**? Does the brain have similar mechanisms?

Future Directions

- Advance and understand, e.g.:
 - Relax! support approximate matching, overlapping active concepts, ..
 - Concept generation and incorporation, interaction with interpretation, etc.
- **Support abstraction, inside the concept structures**
 - Beyond strings: learn more general subclass of finite state machines
- Larger datasets and other modalities:
 - “Scale” to other (perceptual) domains (e.g. audio or images)
 - Multiple signals, more dimensions, additional phenomena...
 - Beyond spatiotemporal... general relations??
- Control of input (attention)

Compositionality/Hierarchies in Computer Vision

Z. Si and S.-C. Zhu (2013). Learning And-Or templates for object recognition and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*

S. Fidler, M Boben, and A. Leonardis. (2014). Learning a Hierarchical Compositional Shape Vocabulary for Multi-class Object Representation. ArXiv.

Zhu, L., Lin, C., Huang, H., Chen, Y., and Yuille, A. (2008). Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidences and competitive exclusion.

Geman, S. (1999). Hierarchy in machine and natural vision. In *Proceedings of the Scandinavian Conference on Image Analysis*.

Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. *Psychological Review*

Thank You!

www.omadani.net

And many thanks to my Cisco managers, Jana Radhkrishinan and Shashi Gandham for the freedom, and to the SAIRG meetings/ members, including Tom Dean, Brian Burns, Reza Eghbali, Gene Lewis, Justin Wang, Akash, Shaunak (for support, great discussions, and valuable feedback).

Extra Slides

What are Interpretation and Segmentation?

- Interpretation (“conceptual analysis”): in an episode, mapping chunks of text (lowest level observations) into highest-level concepts, making sure all text is covered, picking a best mapping (into internal concepts)
- Segmentation: separate or join input lines, group consecutive primitives (with and without concept info), decide to skip some, etc.
- They work closely together

Dataset

- 120k NSF abstracts, 2.5 mil. lines, 20mil term, UCI ml repository
 - (and newsgroups dataset for log-loss test)
- Each episode is a line of text, 55 characters on average
- **Blank spaces removed!** No other preprocessing
- Timings, on Mac Book Pro, window size of 3, beam of 10 & 3:
 - 3 minutes for 1500 lines, up to layer 1 (up to bigrams!)
 - 30 minutes, for 1500 lines, up to layer 4

Approach

- Prediction!
 - But what to predict?? And representations, objectives, etc?
- Build a self-supervised learning system on the text stream
 - (e.g. natural language corpora, the web, tweets, computer logs, ...)
 - Begins at a low level, eg character level, and makes “higher level concepts”
 - Concepts would have associations with one another
 - Concepts would have a hierarchical (recursive) structure
 - Are constantly validated/updated by what is seen (the lowest level..)
- Ample rich data (unlabeled), e.g. text/natural language, a rich hierarchy of structure hidden or regularities:
 - characters → area-code → phone → contact-info → resume → ...
 - characters → words → phrases → expressions → ...

Challenges/Opportunities

- Local decisions/local search.. poor local optima?
- Much noise, yes, but constraints from what is learned (e.g. active concepts) can adequately guide the learning (plus, of course, appropriate algorithms and biases..)
- Non-stationarity
- Multiple interacting learners/inferencers working together, helping one another
- Objective(s)?



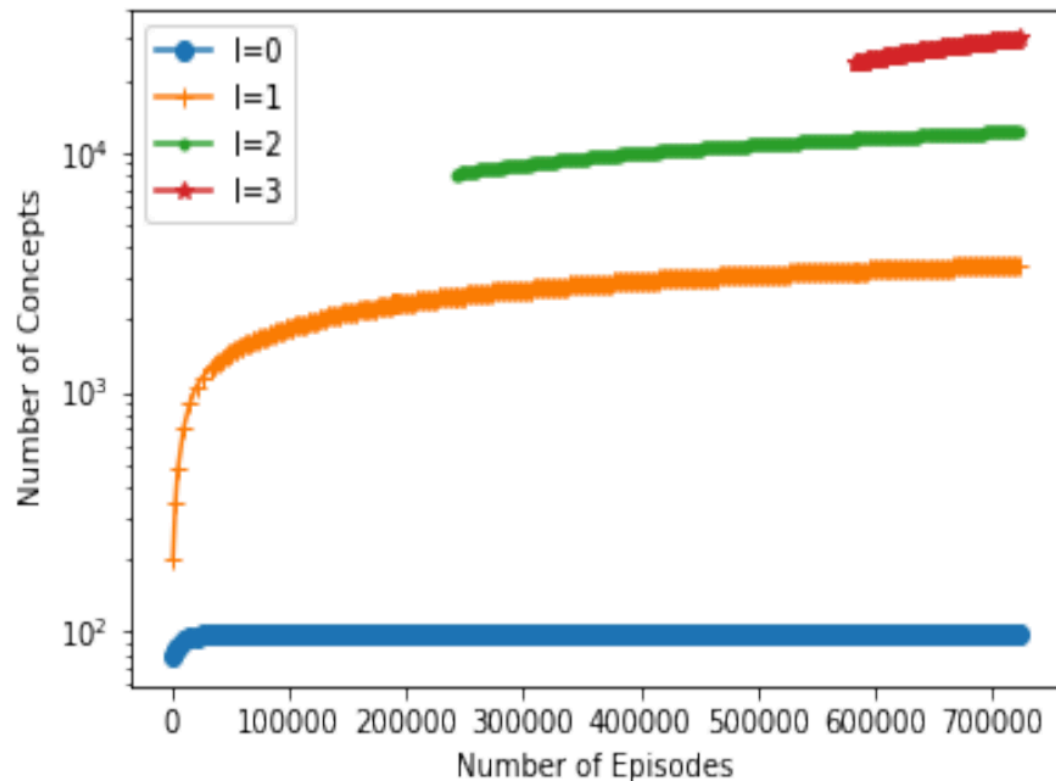
Example Concepts

concept	freq.	last seen	score*	concept	freq.	last seen	score
con_2 ('ther')	57890	25	14.0	con_3 ('sand')	54456	24	10.1
con_3 ('ther')	6370	106	8.0	con_3 ('research')	50353	20	25.3
con_3 ('there')	4023	58	10.4	con_3 ('project')	42501	22	28.0
con_2 ('with')	84643	16	17.1	con_3 ('ation')	36479	101	13.4
con_3 ('with')	22195	48	10.8	con_3 ('develop')	28092	85	25.5
con_3 ('whether')	3383	388	21.0	con_3 ('s')	966729	2	1.1

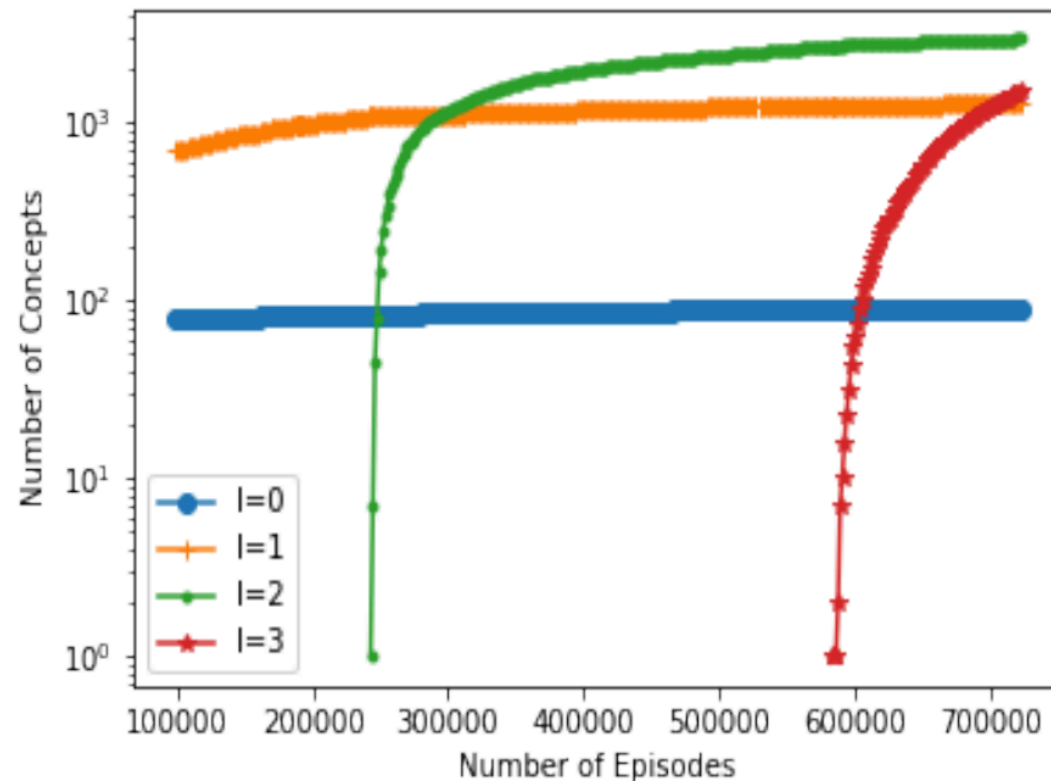
Components of Learning and Inference

- Learning:
 - Updating prediction (association) weights (from co-occurrences)
 - Updating other concept statistics: seen-count, reward (moving averages, etc)
 - Composing to create bigrams/holonyms, and adding a new layer
- Inference:
 - Interpretation: semantics, data structures, algorithm(s)

Growth in Number of Concepts



(a) Number of concepts observed.



(b) Number of non-clone concepts with frequency ≥ 100 .

Figure 8: Number of unique concepts observed (in a segmentation) for each level, $l = 0, 1, \dots$, vs. time (number of episodes, or lines read), during training of Model4, up to episode 700k.

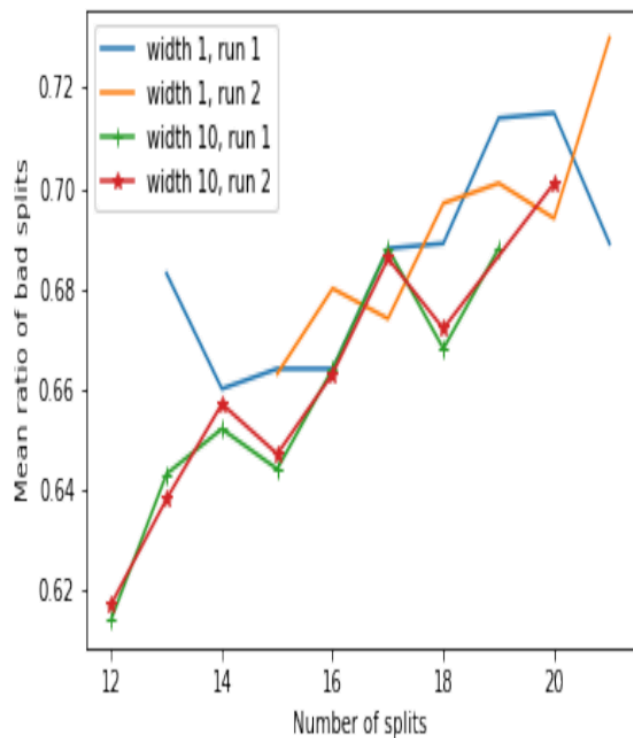
Bad Splits and Bad Ratios (Go down with training and/or inference)

	level 1 model			level 3 model (Model3)			level 4 model (Model4)		
try, keep	score	bad1	bad2	score	bad1	bad2	score	bad1	bad2
1,1	1.64	23.5	17.2	6.9	11.3	6.7	5.6	10.6	6.6
2,2	1.77	23.3	17.1	8.0	10.5	6.1	7.3	9.3	5.5
5,5	1.89	23.1	16.8	8.96	9.8	5.4	10.6	7.3	4.1

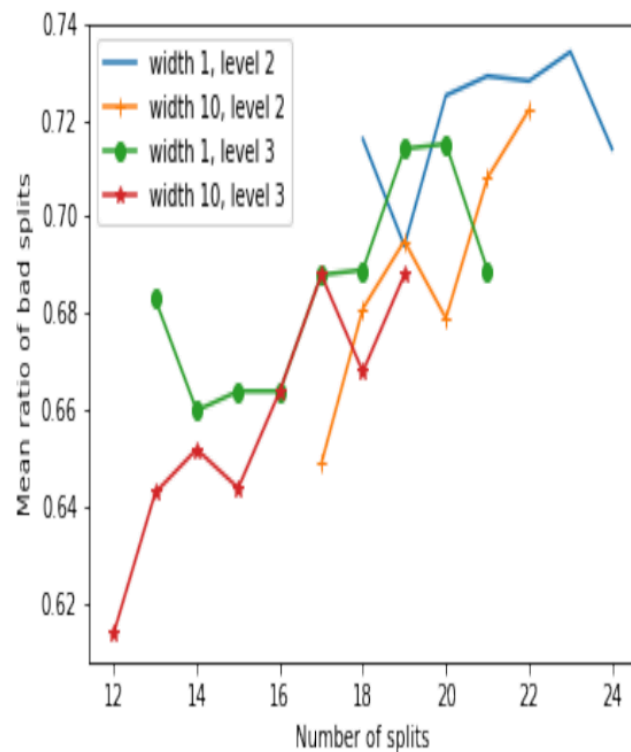
Table 9: Average COMA and average count of bad splits (over 188 episodes) for several levels and beam search parameters. As expected, COMA improves with additional search. Number of bad splits and COMA are (negatively) correlated too.

level 2 model		Model3		
width 1,1	width 10,10	width 1,1	width 3,3	width 10,10
0.735	0.702	0.701	0.687	0.672

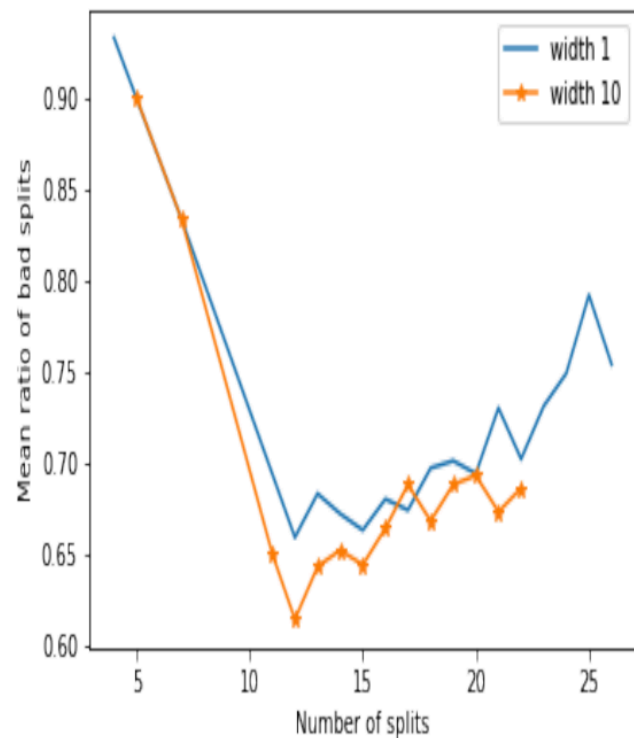
Table 10: Bad-split ratio scores for a few models and segmentation beam widths, average over the same 388 episodes. The bad ratios decrease with more training and additional inference (width). Multiple runs gives similar results. See also Fig. 14.



(a) Model3: 2 runs with width 1, 2 runs with width 10 (*i.e.* 10, 10).



(b) Model3 and a model trained up to level 2.



(c) Model3: Minimum requirement of 5 episodes.

Figure 14: Plots of averages of bad-split ratios for those split counts where we got at least 20 episodes, (a) and (b), or minimum of 5 for (c), for the total split count, when a model was run over nearly 400 episodes. As we increase the beam width or the training, the plots move to the left and the bad-split ratios decrease.

pos=-1	pos=1	pos=2	pos=-1	pos=1	pos=2
<i>con₀</i> ('a') ('a' at level 0)			<i>con₃</i> ('a') ('a' at level 3)		
"r",0.107	"n",0.200	"i",0.155	"s",0.076	"s",0.091	"a",0.034
"e",0.101	"t",0.191	"e",0.146	"t",0.063	"te",0.025	"s",0.034
"t",0.079	"l",0.143	"d",0.125	"n",0.059	"in",0.018	"e",0.025
"n",0.074	"r",0.120	"t",0.087	"dat",0.038	"i",0.018	"t",0.021
"c",0.073	"s",0.065	"a",0.064	"w",0.037	"c",0.017	"n",0.019
"h",0.070	"c",0.059	"o",0.050	"e",0.029	"d",0.016	"o",0.015
"l",0.070	"b",0.029	"c",0.048	"te",0.022	"ble",0.014	"i",0.012
"m",0.065	"d",0.027	"s",0.047	"an",0.017	"nt",0.013	"to",0.012
"s",0.057	"m",0.027	"l",0.044	"to",0.016	"re",0.011	"the",0.011
"p",0.034	"p",0.025	"n",0.026	"in",0.016	"ch",0.009	"d",0.010
<i>con₁</i> ('th')			<i>con₃</i> ('th')		
"wi",0.215	"er",0.146	"t",0.103	"e",0.066	"at",0.287	"the",0.039
"o",0.072	"a",0.143	"u",0.073	"o",0.053	"is",0.052	"a",0.038
"me",0.067	"o",0.125	"s",0.070	"dep",0.043	"s",0.047	"s",0.034
"bo",0.039	"e",0.122	"d",0.069	"Sou",0.038	"us",0.037	"e",0.024
"ng",0.034	"t",0.053	"e",0.059	"a",0.037	"an",0.035	"t",0.019
"a",0.032	"i",0.049	"n",0.049	"pa",0.034	"n",0.028	"ca",0.014
"i",0.032	"ro",0.046	"he",0.048	"veleng",0.021	"erin",0.018	"f",0.013
"e",0.029	"re",0.037	"a",0.042	"algori",0.018	"no",0.018	"n",0.013
"or",0.027	"m",0.023	"i",0.031	"workwi",0.018	"r",0.017	"ust",0.013
"d",0.027	"s",0.021	"se",0.027	"leng",0.017	"in",0.015	"ic",0.012
<i>con₂</i> ('ther')			<i>con₃</i> ('ther')		
"o",0.311	"e",0.100	"a",0.050	"i",0.168	"mody",0.058	"n",0.076
"whe",0.076	"s",0.067	"e",0.038	"wea",0.096	"es",0.028	"e",0.042
"ge",0.063	"mo",0.050	"s",0.037	"ando",0.086	"mos",0.022	"a",0.034
"fur",0.050	"n",0.048	"n",0.035	",ando",0.066	"p",0.019	"s",0.025
"u",0.036	"t",0.042	"t",0.032	"so",0.041	"a",0.018	"t",0.025
"i",0.033	"the",0.036	"o",0.026	"Wea",0.026	"mal",0.018	"o",0.020
"a",0.030	"mal",0.029	"i",0.021	"e",0.022	"b",0.017	"i",0.015
".Fur",0.026	"i",0.027	"se",0.019	"O",0.020	"pre",0.013	"y",0.014
"ro",0.025	"th",0.024	"dy",0.019	"ur",0.019	"lands",0.012	"p",0.013
"ra",0.025	"more",0.021	"g",0.014	"sei",0.019	"to",0.012	"te",0.011

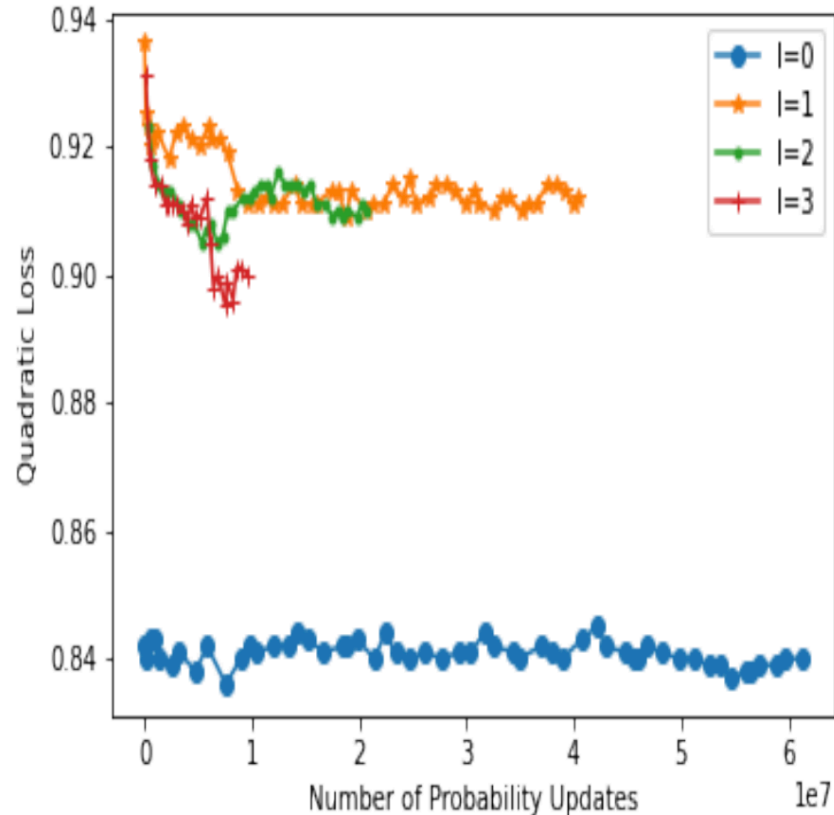
Top 10 prediction edges for a few concepts.

Number of Prediction Edges and Prob. Mass

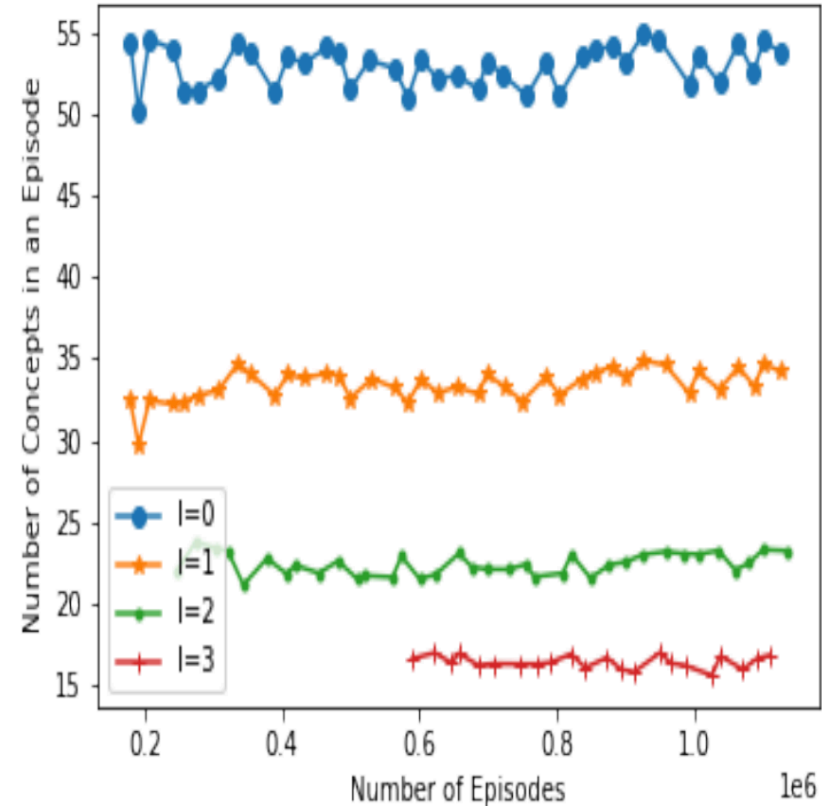
	$p \geq 0.0$	$p \geq 0.01$		$p \geq 0.0$	$p \geq 0.01$		$p \geq 0.10$			
	freq. ≥ 50			freq. ≥ 100			freq. ≥ 50		freq. ≥ 100	
level	pos=1	pos=1	pos=2	pos=1	pos=1	pos=2	pos=1	pos=2	pos=1	pos=2
3	87	0.80, 16.6	0.64	97	0.78, 16.5	0.59	0.41, 1.46	0.15	0.39, 1.4	0.13
2	129	0.81, 15.2	0.65	134	0.80, 15.0	0.63	0.44, 1.6	0.2	0.44, 1.6	0.2
1	112	0.84, 20.6	0.78	139	0.79, 18.9	0.72	0.31, 1.44	0.16	0.32, 1.46	0.16
0	70	0.90, 17.2	0.89	70	0.90, 17.3	0.89	0.44, 2.3	0.28	0.44, 2.3	0.28

Table 6: Average (over concepts) of edge-weight (probability) mass over prediction edges with weight surpassing 0.01 and 0.1 thresholds for pos=1 and pos=2. The average number of such edges is also shown for pos=1, and thresholds 0 (no threshold), 0.01 and 0.1. Thus at level 3, for concepts with freq. ≥ 50 , on average there are 87 edges per concept, and about 17 edges for such concepts have a weight of no less than 0.01, and this average is 1.4 for thresh. 0.1 (all for pos=1). We observe the number of edges goes up with growing concept freq. (from 50 to 100) but the number of edges with high weight can slightly go down. Across layers, the total mass of edges does not change much, and much of the mass, around 80% or more, are on edges with weight ≥ 0.01 for pos=1.

Quadratic Loss and Num Concepts per Episode



(a) Squared loss on probabilities



(b) Number of concepts per episode, at each level.

Figure 13: Progress in quadratic loss and average number of concepts in in each level of the selected chain per episode, over time. The number of primitives (level 0 concepts) is about 54 per episode. The convergence is fast and these measures are only crude ways of measuring system progress, compared to the COMA score over time and for various levels (Figs. 12 and 11).

System Overview

Learning

Update Prediction Edges

(create/weight-update/discard edges)

Update Concept Statistics

(priors, scores, ..)

Compose Concepts

(create new concepts, via concatenating,
add part-edges of the hierarchy)

the network
→
(concepts + edges)

(co-)occurrences
←

Inference

Segmentation + Interpretation

(*making sense* of the input:
producing what&where,
involving predicting, matching,
search, exploration&exploitation)

Binary Sequence Experiments: 0 or 1 Primitives!

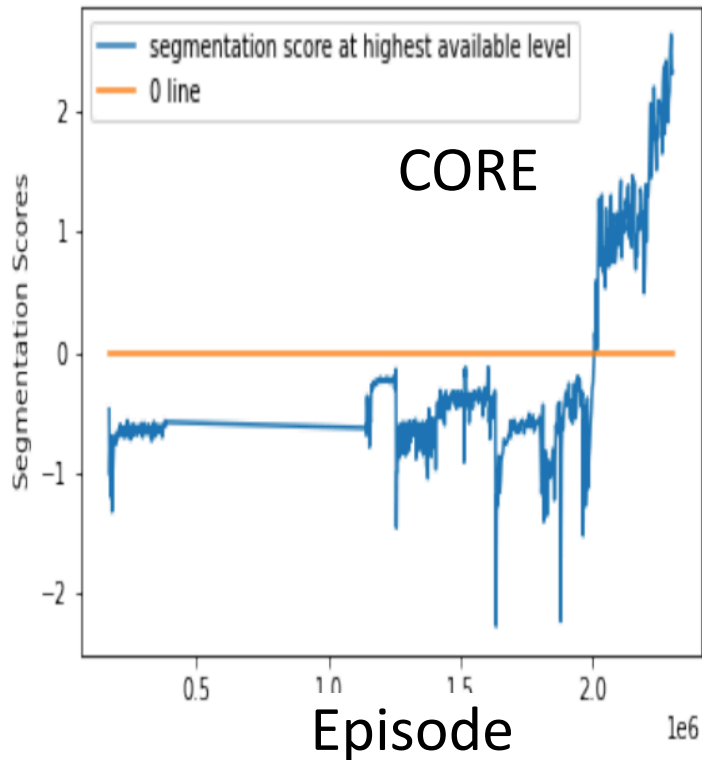
- Convert each character to an 8-bit code

'g'='00000000'	'o'='00000001'	'a'='00000010'	'l'='00000010'	't'='00000100'	'd'='00000101'
'e'='00000110'	's'='00000111'	'c'='00001000'	...	'I'='00100111'	'('='00101001'

Table 13: The binary encoding of a few characters, so the word "goal" becomes the string "00000000000000001000000100000010" input to the system.

Progress on the Binary Stream (CORE, etc)

- Initially, negative average CORE for several levels! But then semantic progress!



Num. concepts per episode

