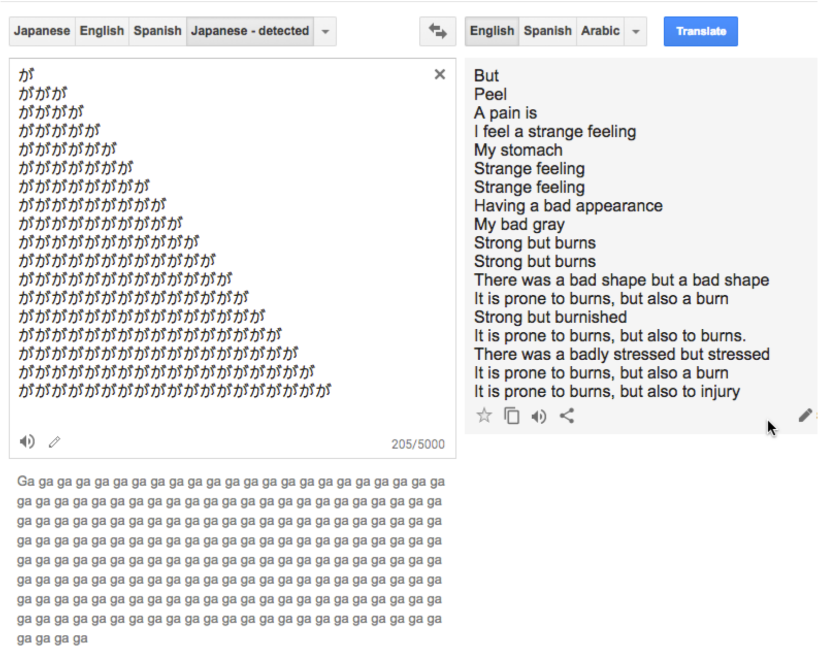
Learning with Partially Ordered Representations

Jonathan Rawski
Department of Linguistics
IACS Research Award Presentation

August 11, 2018

The Main Idea

Learning is eased when attributes of elements of sequences structure the space of hypotheses


Japanese
English
Spanish
Japanese - detected
English
Spanish
Arabic
Translate
が
ががが
がががが
ががががが
がががががが
ががががががが
がががががががが
ががががががががが
がががががががががが
ががががががががががが
がががががががががががが
ががががががががががががが
がががががががががががががが
ががががががががががががががが
がががががががががががががががが
ががががががががががががががががが
がががががががががががががががががが
ががががががががががががががががががが
205/5000
Ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga
ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga
ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga
ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga
ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga
ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga
ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga
ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga ga
ga ga ga ga
But
Peel
A pain is
I feel a strange feeling
My stomach
Strange feeling
Strange feeling
Having a bad appearance
My bad gray
Strong but burns
Strong but burns
There was a bad shape but a bad shape
It is prone to burns, but also a burn
Strong but burnished
It is prone to burns, but also to burns.
There was a badly stressed but stressed
It is prone to burns, but also a burn
It is prone to burns, but also to injury

Poverty of the Stimulus and Data Sparsity

Number of English words: $\sim$ 10,000
Possible English 2-grams: N^2
Possible English 3-grams: N^3
Possible English 4-grams: N^4
...
easy learning if normal distribution

Poverty of the Stimulus and Data Sparsity

BUT:

In the million-word Brown corpus of English:
45% of words,
80% of 2-grams
95% of 3-grams
appear EXACTLY ONCE
Bad for learning: Huge long-tailed distribution

How can a machine know that new sentences like
"nine and a half turtles yodeled" is good?
"turtles half nine a the yodeled" is bad?

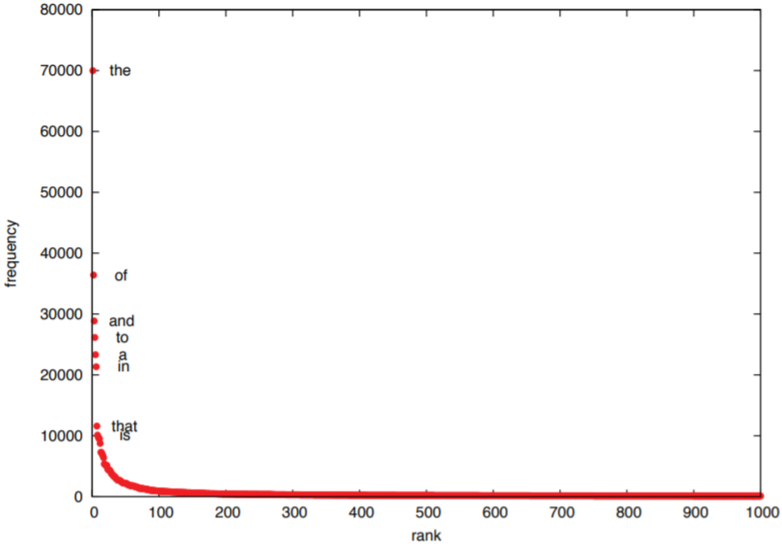
Poverty of the Stimulus and Data Sparsity

BUT:

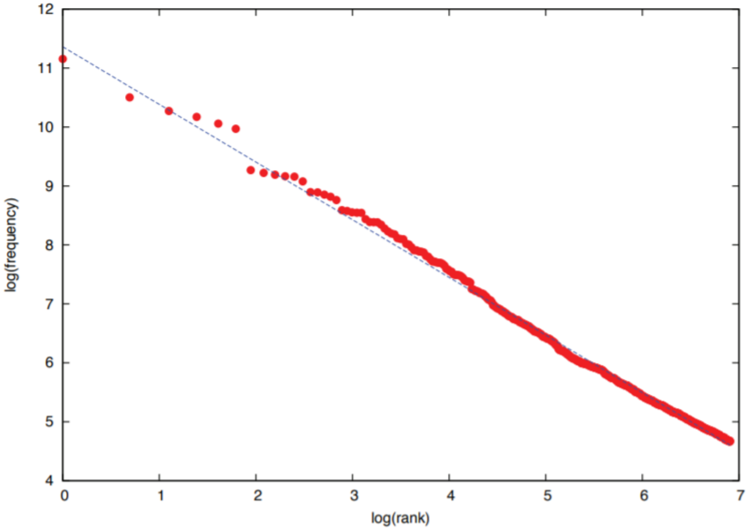
In the million-word Brown corpus of English:
45% of words,
80% of 2-grams
95% of 3-grams
appear EXACTLY ONCE
Bad for learning: Huge long-tailed distribution

How can a machine know that new sentences like
"nine and a half turtles yodeled" is good?
"turtles half nine a the yodeled" is bad?

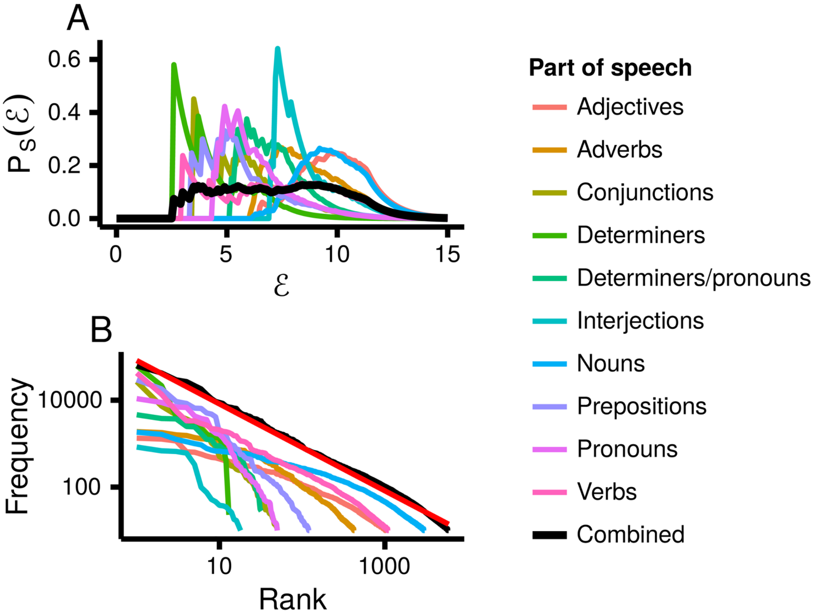
The Zipf Problem



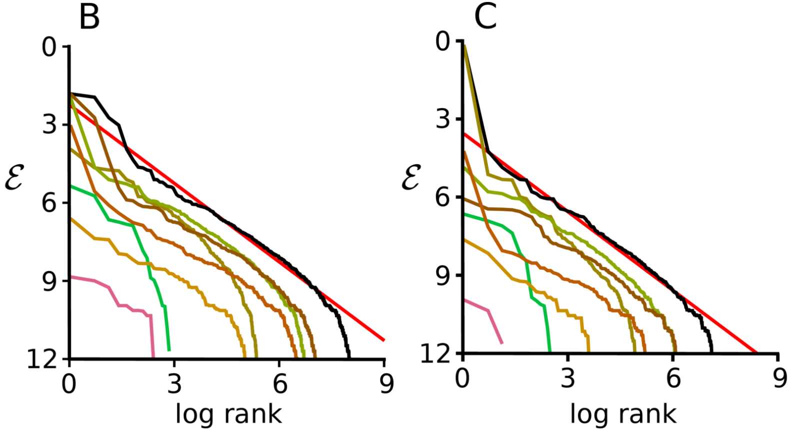
The Zipf Problem



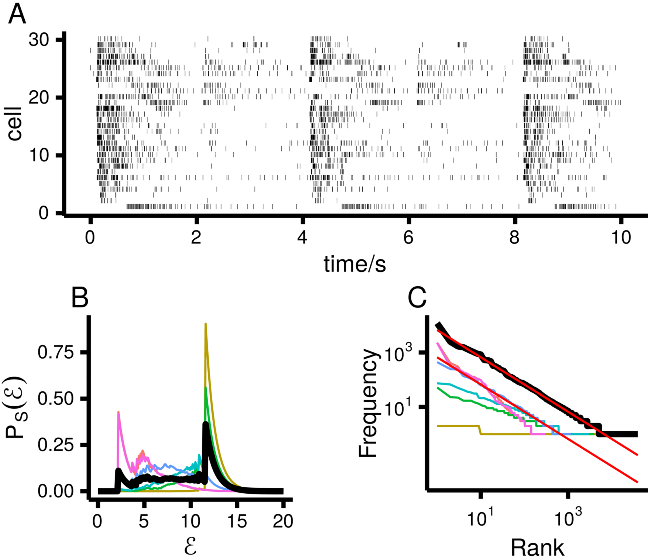
Zipf Emerges from Latent Features



Zipf Emerges from Latent Features



Zipf Emerges from Latent Features

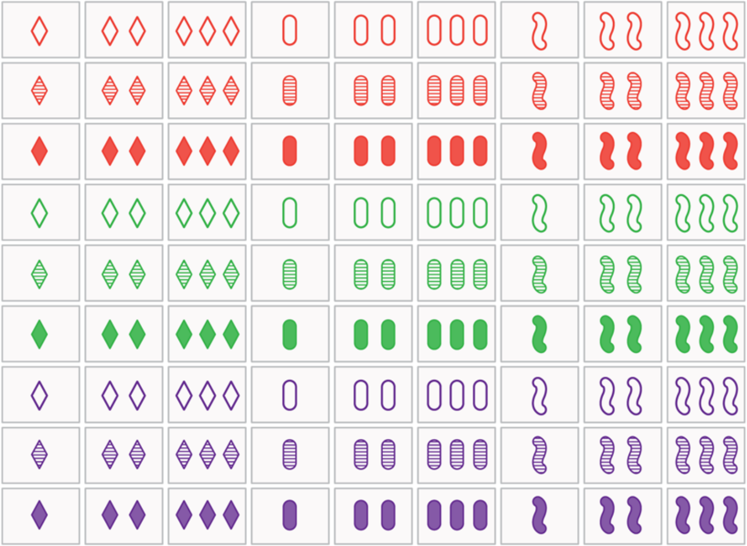


SET OR NO SET

Some examples below:

Are the attributes all the same or all different?				
Color	✖	✖	all different	all the same
Shape	all different	all different	all different	all the same
Shading	all different	all different	all different	all the same
Number	✖	all different	all different	all different
	Not a set	Not a set	A set	A set

THE FULL DECK



Learning Algorithm (Chandlee et al 2018)

What have we done so far?

- Provably correct relational learning algorithm
- Prunes Hypothesis space according to ordering relation
- Provably identifies correct constraints for sequential data
- Uses data sparsity to its advantage!

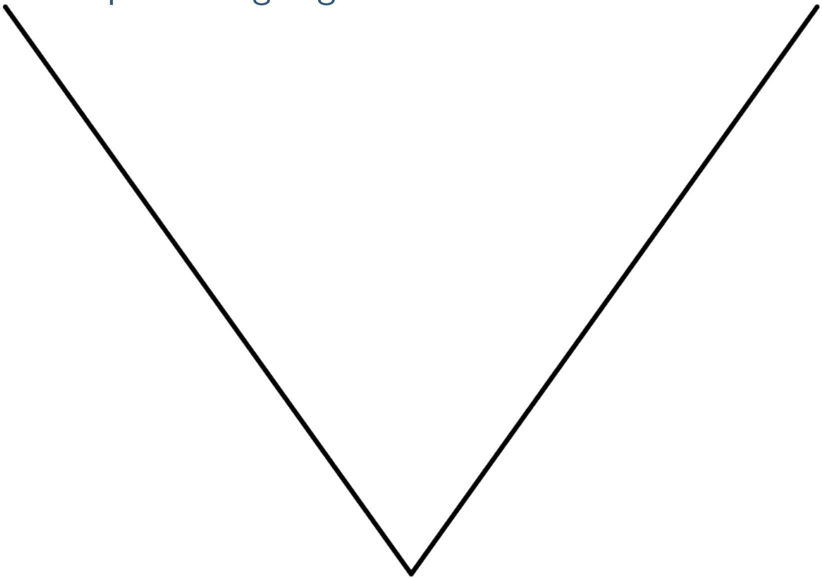
Collaborative work with:

Jane Chandlee
(Haverford)

Jeff Heinz
(SBU)

Adam Jardine
(Rutgers)

Bottom-Up Learning Algorithm



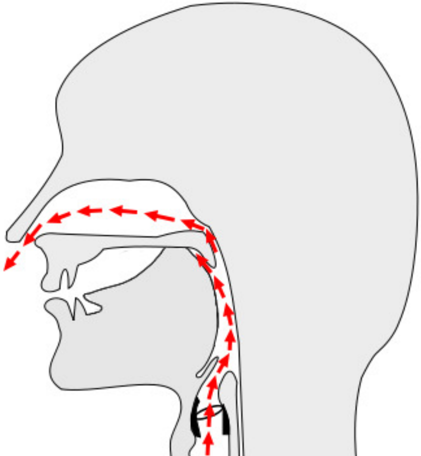
Example: Features in Linguistics

sing

ring

bling

ng $=$ [$+$Nasal,$+$Voice,$+$Velar]



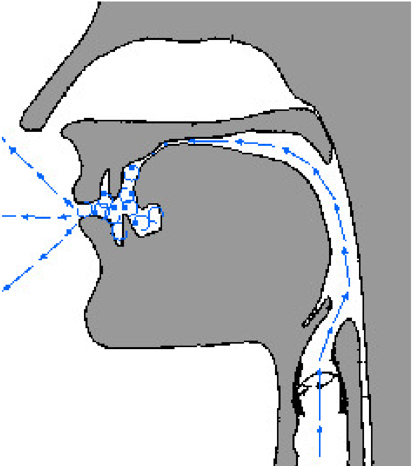
Example: Features in Linguistics

sand

sit

cats

s= [-Nasal,-**Voice**,- **Velar**]

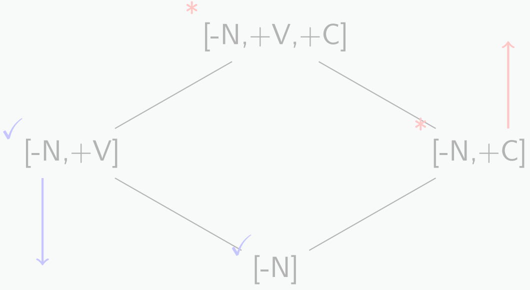


Structuring the Hypothesis Space: Feature Matrix Ideals

Feature Inventory

- $\pm$N = Nasal
- $\pm$V = Voiced
- $\pm$C = Consonant

Example

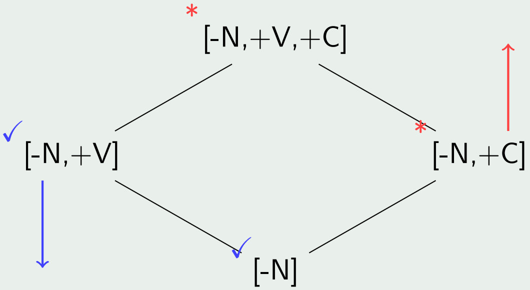


Structuring the Hypothesis Space: Feature Matrix Ideals

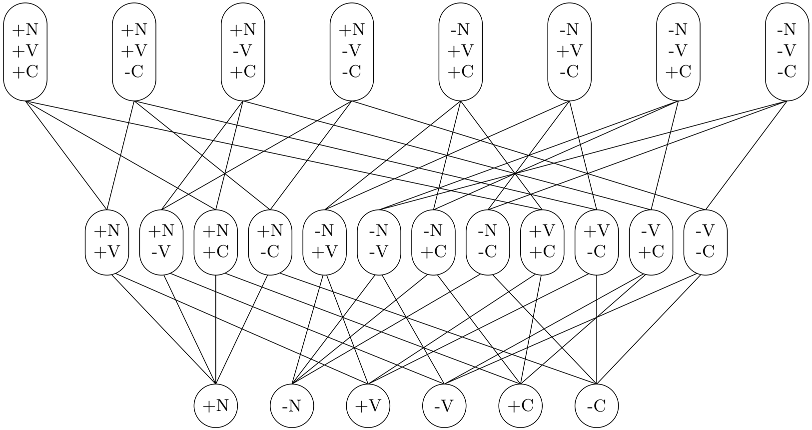
Feature Inventory

- $\pm$N = Nasal
- $\pm$V = Voiced
- $\pm$C = Consonant

Example

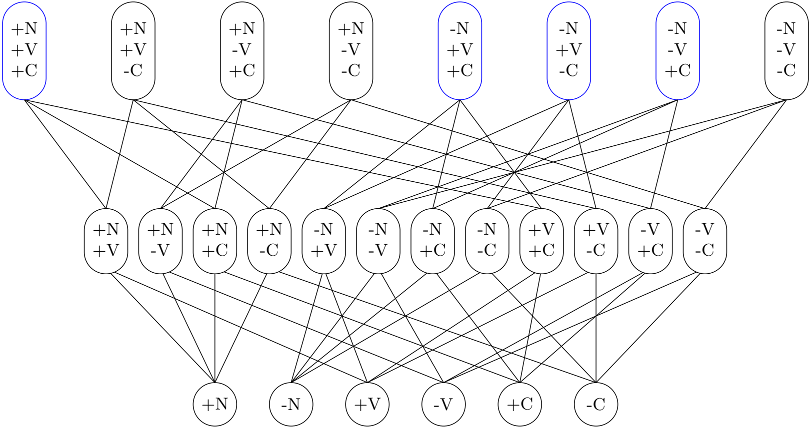


Example



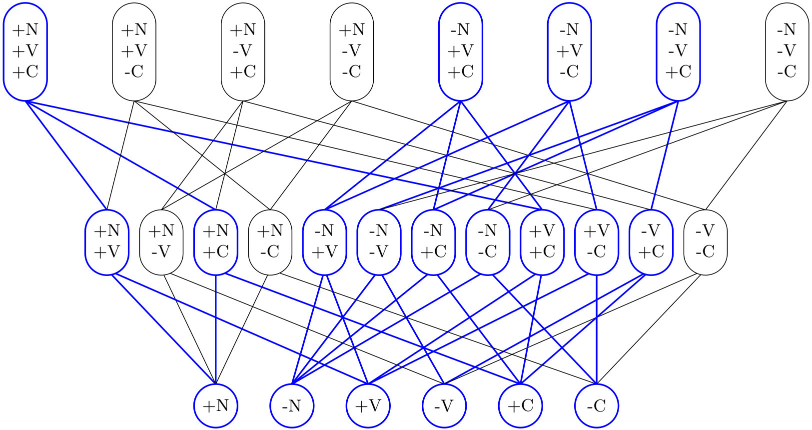
+N +V +C
+N +V -C
+N -V +C
+N -V -C
-N +V +C
-N +V -C
-N -V +C
-N -V -C
+N +V
+N -V
+N +C
+N -C
-N +V
-N -V
-N +C
-N -C
+V +C
+V -C
-V +C
-V -C
+N
-N
+V
-V
+C
-C

Example



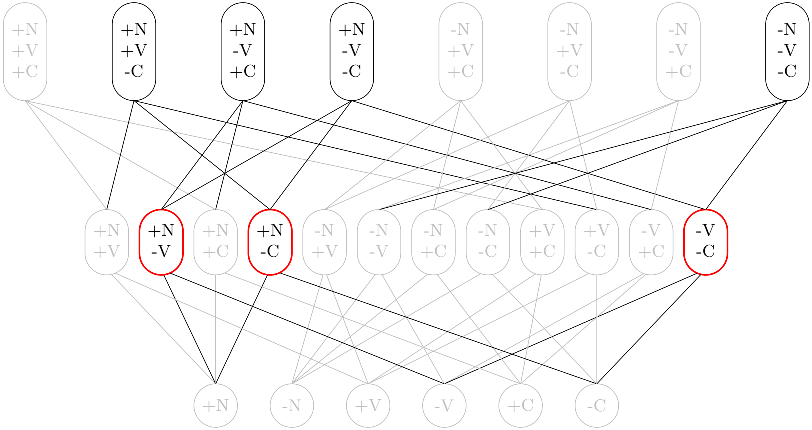
+N +V +C
+N +V -C
+N -V +C
+N -V -C
-N +V +C
-N +V -C
-N -V +C
-N -V -C
+N +V
+N -V
+N +C
+N -C
-N +V
-N -V
-N +C
-N -C
+V +C
+V -C
-V +C
-V -C
+N
-N
+V
-V
+C
-C

Example



+N +V +C
+N +V -C
+N -V +C
+N -V -C
-N +V +C
-N +V -C
-N -V +C
-N -V -C
+N +V
+N -V
+N +C
+N -C
-N +V
-N -V
-N +C
-N -C
+V +C
+V -C
-V +C
-V -C
+N
-N
+V
-V
+C
-C

Example


+N +V +C
+N +V -C
+N -V +C
+N -V -C
-N +V +C
-N +V -C
-N -V +C
-N -V -C
+N +V
+N -V
+N +C
+N -C
-N +V
-N -V
-N +C
-N -C
+V +C
+V -C
-V +C
-V -C
+N
-N
+V
-V
+C
-C

Two Ways to Explore the Space

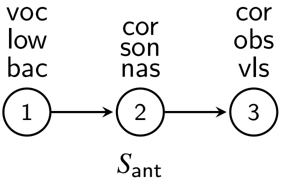
Top-Down Induction

- Start at the most specific points (highest) in the semilattice
- Remove all the substructures from the lattice that are present in the data.
- Collect the most general substructures remaining.

Bottom-Up Induction

- Beginning at the lowest element in the semilattice,
- Check whether this structure is present in the input data.
- If so, move up the lattice, either to a point with an adjacent underspecified segment, or a feature extension of a current segment, and repeat.

Semilattice Explosion



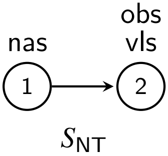
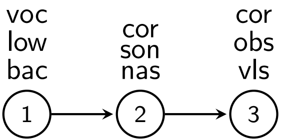


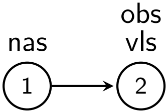
Table 2
Number of possible constraints for various values of $|C|$ and n

		$|C|$			
		30	**100**	**200**	**400**
	1	30	100	200	400
	2	900	10,000	40,000	160,000
n	**3**	27,000	1,000,000	8 million	64 million
	4	810,000	100 million	1.6 billion	26 billion
	5	24 million	10 billion	320 billion	10 trillion

Semilattice Explosion



S_{ant}



S_{NT}

Table 2
Number of possible constraints for various values of $|C|$ and n

		$\vert C\vert$			
		30	**100**	**200**	**400**
	1	30	100	200	400
	2	900	10,000	40,000	160,000
n	**3**	27,000	1,000,000	8 million	64 million
	4	810,000	100 million	1.6 billion	26 billion
	5	24 million	10 billion	320 billion	10 trillion

Plan of the project

What has been done

Provably correct bottom-up learning algorithm

Goals of the Project

- Model Efficiency
- Model Implementation
- Model Testing - large linguistic datasets
- Model Comparison: UCLA Maximum Entropy Learner

Broader Impacts

- Learner that takes advantage of data sparsity
- applicable on any sequential data (language, genetics, robotic planning, etc.)
- implemented, open-source code

Project Timeline 2018-2019

Month	**Plan**
September October	Algorithmic Efficiency Implement string-to-model functions in Haskell
November December January	Implement top-down learner in Python3 Implement bottom-up learner in Python3
Febuary March	test learning algorithm - Brazilian Quechua corpus
April May	Model Comparison with Maximum Entropy Learner & Deep Networks

future work	Extend from learning patterns to transformations test on other linguistic sequence data (syntax) extend to other non-linguistic sequences extend to robotic planning

The Main Idea

Learning is eased when attributes of elements of sequences structure the space of hypotheses

Lila Gleitman (1990)

"the trouble is that an observer who notices *everything* can learn *nothing*, for there is no end of categories known and constructable to describe a situation"