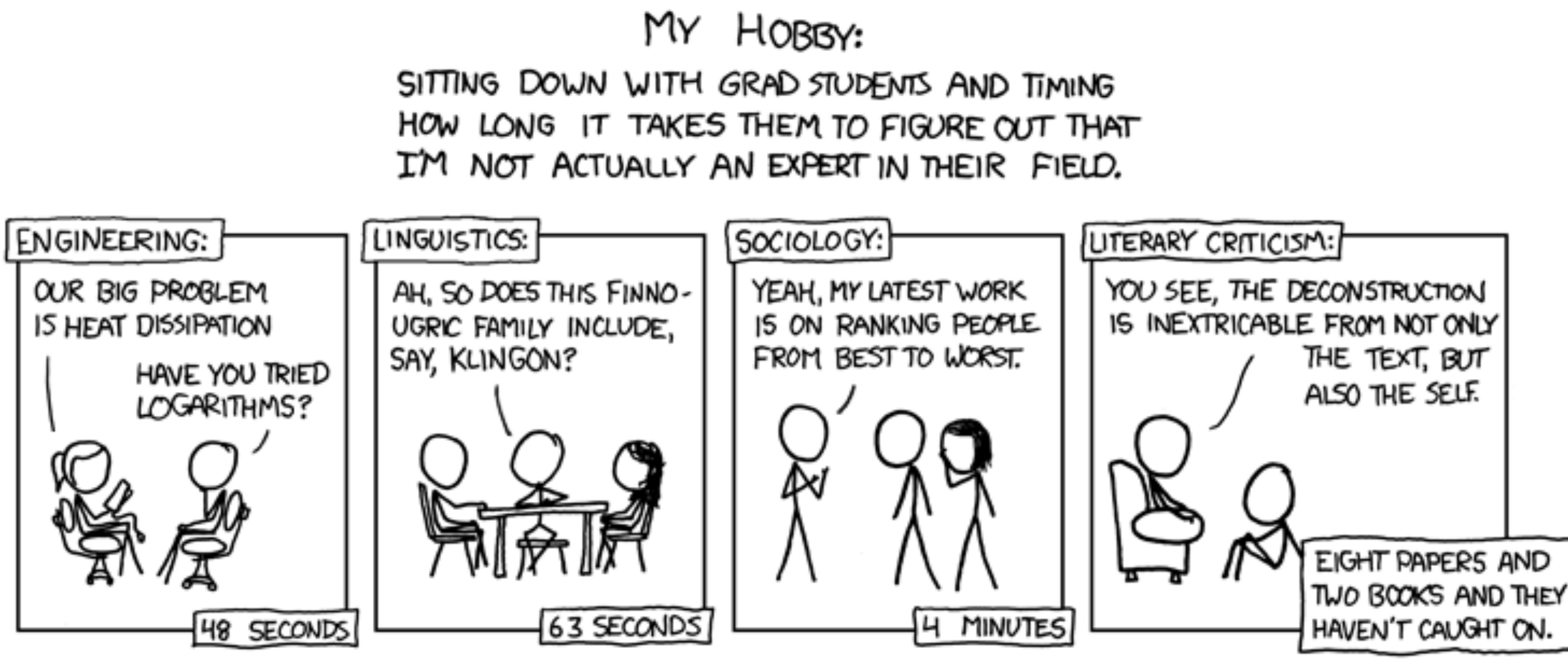
Computational linguistics



(xkcd)

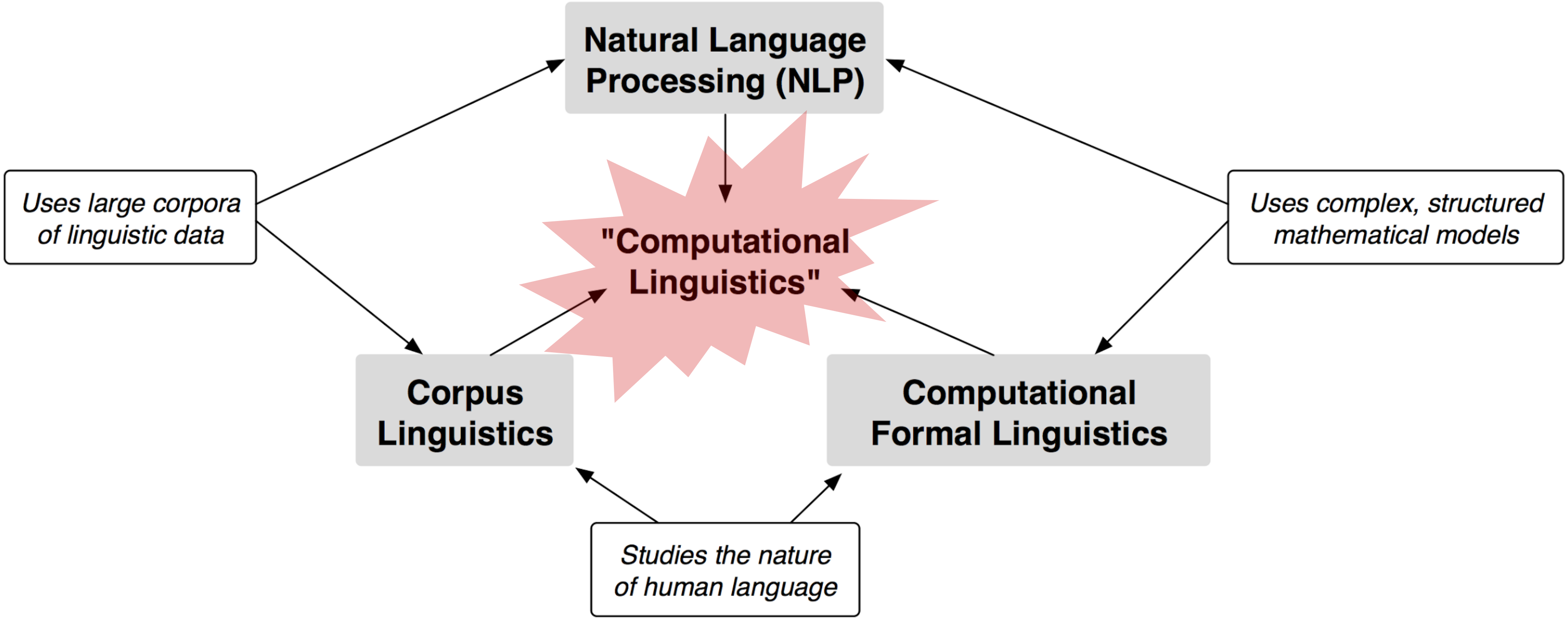
You are here.

Sam Bowman

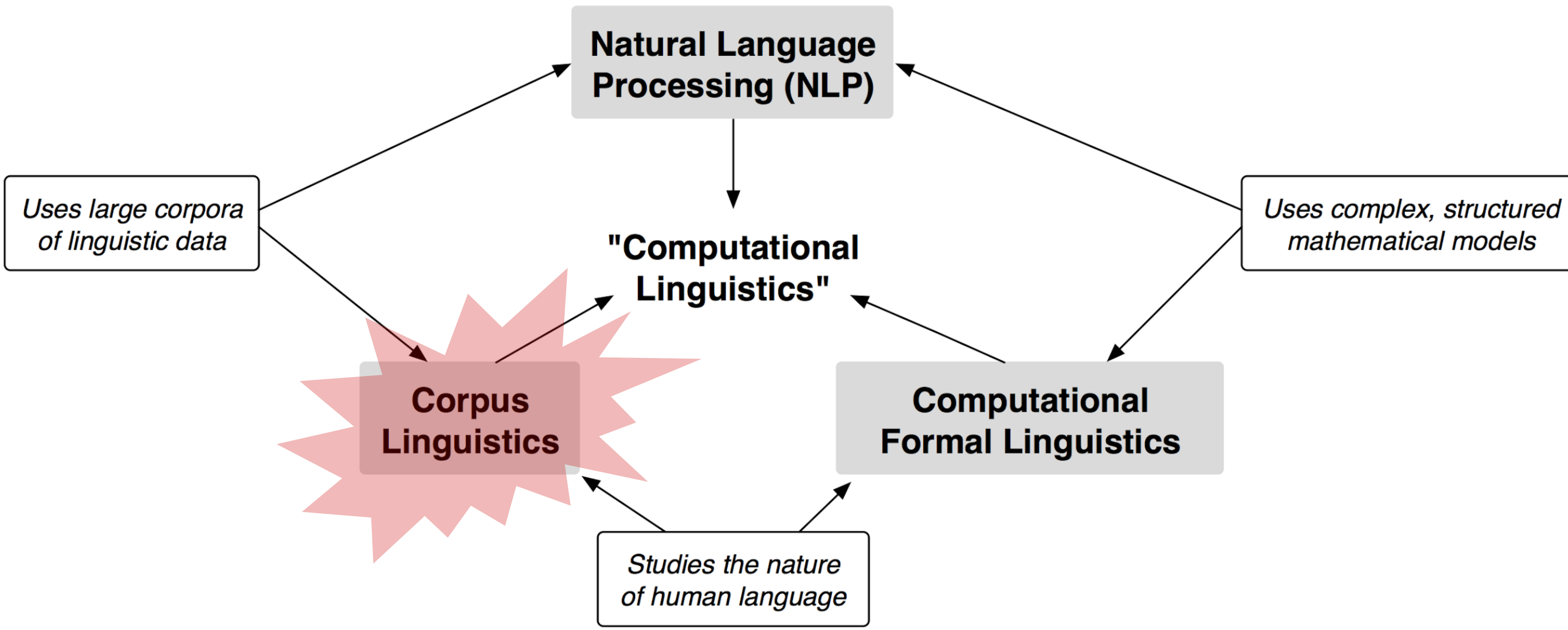
PhD student, Linguistics/NLP Group

What is computational linguistics?

What is computational linguistics?



What is computational linguistics?



Corpus linguistics

- Uses observations from large collections of naturally occurring language data, called *corpora* (singular: *corpus*), to answer questions about human language.

why?

Is it a corpus?

- The complete archive of the New York Times?
- Ten audio interviews with strangers at the Caltrain station, with transcripts?
- All of Battlestar Galactica on DVD?
- This lecture?

Some frequenly used corpora

- **Brown Corpus** (1964): 1m words, multiple genres
- **Switchboard**: Recorded phone calls between strangers, detailed transcriptions.
- **The Penn Treebank**: Text from both of the above, with parse trees.
- **The Google Web Treebank**: Text drawn from all over the modern English web, with parse trees.
- **CHILDES**: Conversations between parents and young children of various ages.
- **Europarl:** The (spoken) proceedings of the EU parliament, translated line-by-line into the official languages of all of the EU countries.

A semantic corpus study: *need to*

- *Need to* vs. its competitors *have to, got to* (focusing only on obligation, not inference readings; thus ignoring *must*)

- Long observed: *need to* has a subtly different meaning than *have to/got to* – “internal compulsion”
  (1) You **have to** wonder what they were thinking.
  (2) You’ve **gotta** wonder what they were thinking.
  (3) You **need to** wonder what they were thinking.

Work and some slides from [Lelia Glass, NWAV 2014]

Hypothesis

Based on *need to*'s unique semantics, we predict that people are more likely to use you need to when they are in a position to know what would be good for the hearer – when they are experts on the relevant domain; when they play a mentoring role in the hearer's life; or when they are in a position of authority over him.

CHILDES vs. Spoken CoCA

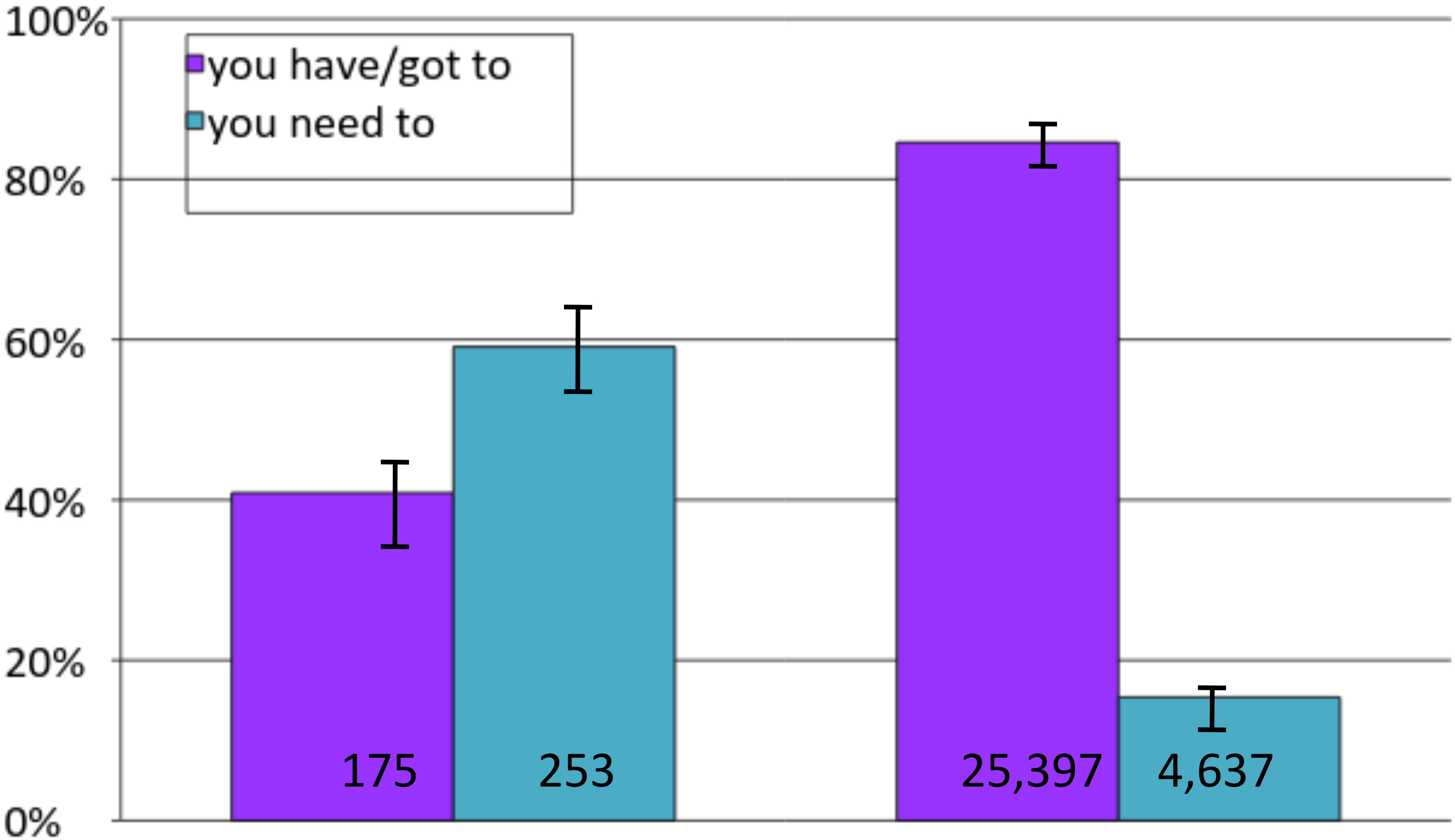
- Prediction: parents will use a higher rate of *you need to* in comparison to *have to/got to* than the general population

- Providence section of CHILDES vs. Spoken CoCA

- (CHILDES: caregiver/child interactions; Spoken CoCA: mostly talk radio shows)

CoCA: Davies 2008;
Providence CHILDES: Demuth, Culbertson & Alter, 2006

Prediction: Parents will use a higher proportion of *you need to* because they (think they) know what's good for their children; occupy mentoring role in their lives



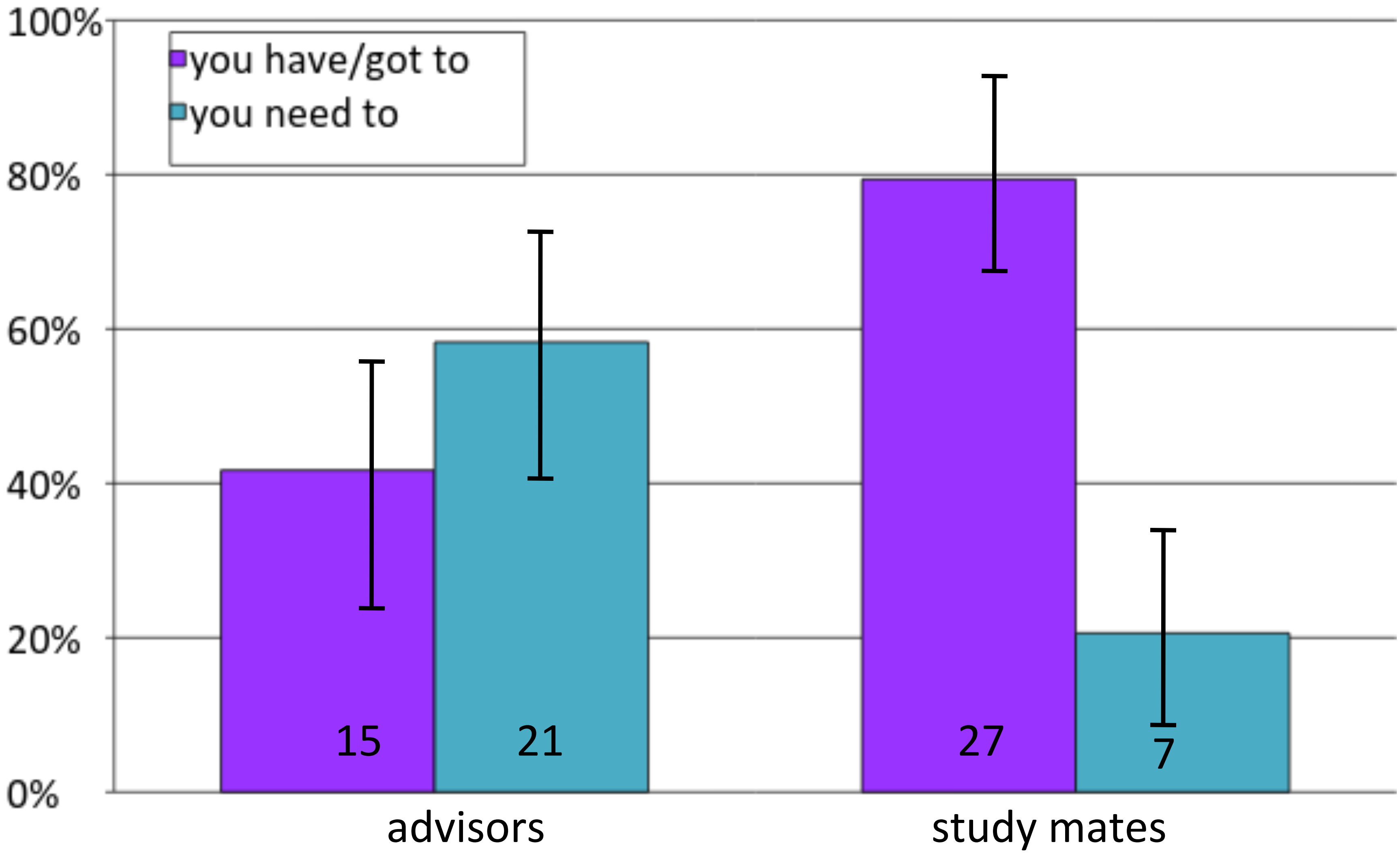
Work and some slides from [Lelia Glass, NWAV 2014]

MiCASE: advisors vs. studymates

- Prediction: academic advisors will use a higher rate of *you need to* in comparison to *have to/got to* than studymates in peer study groups

- Advisors in advising sentences vs. studymates in peer study groups

Prediction: Advisors will use a higher proportion of *you need to* because they have expertise about the subject matter and they are institutionally charged with students' well-being



Stack Exchange Users in Stanford Politeness Corpus

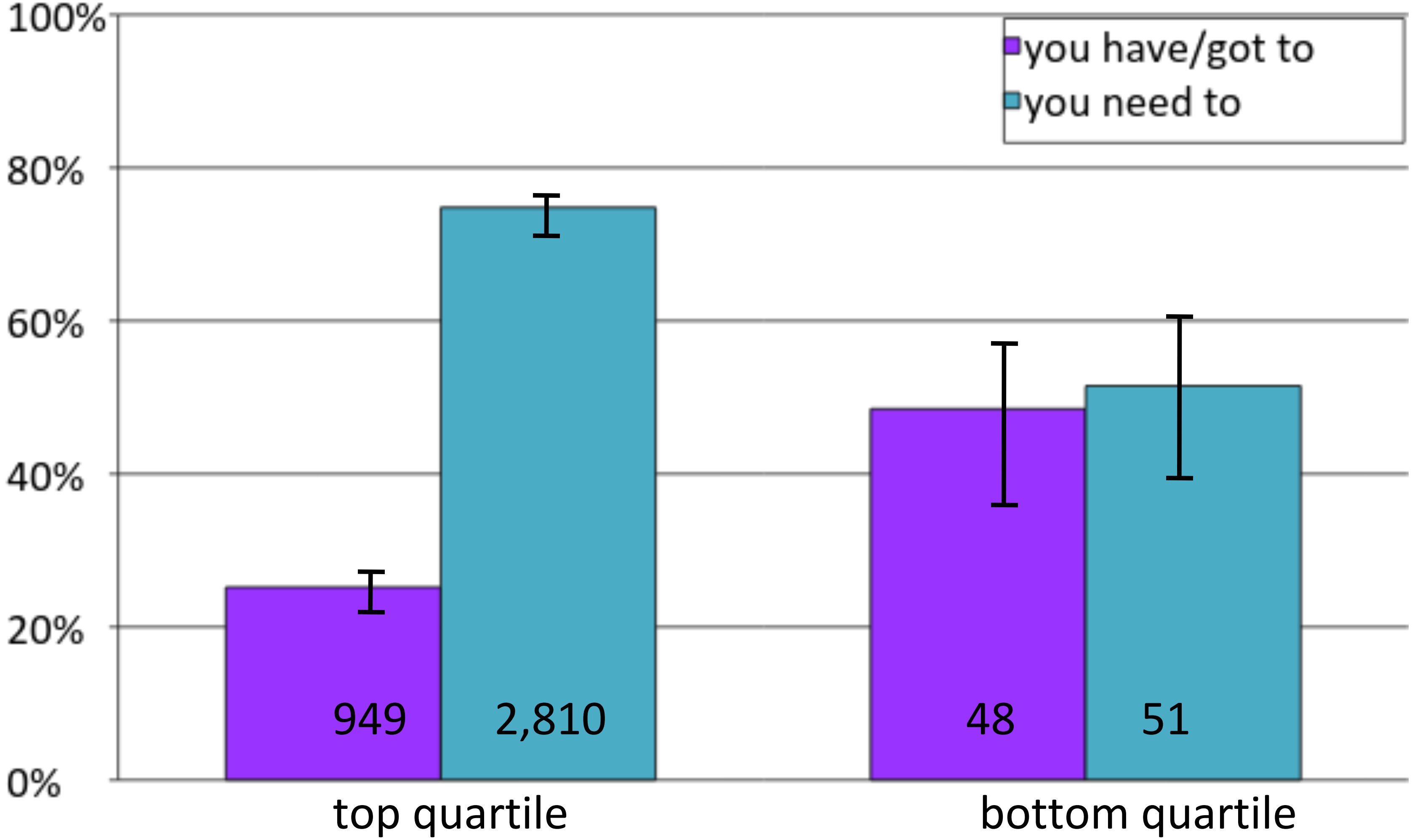
- Highly-rated users are more knowledgeable about subject matter, more qualified to know what would be in the interest of others

- Prediction: highly-rated users will use a higher rate of *you need to* in comparison to *have to/got to* than lower-rated users

Stanford Politeness Corpus: Danescu-Niculescu-Mizil, Sudhof, Jurafsky, Leskovec & Potts 2013

Work and some slides from [Lelia Glass, NWAV 2014]

Prediction: Top-rated users will use a higher proportion of *you need to* because they know what would be good for their addressees (**also significant in logistic regression)



Dwight in *The Office*

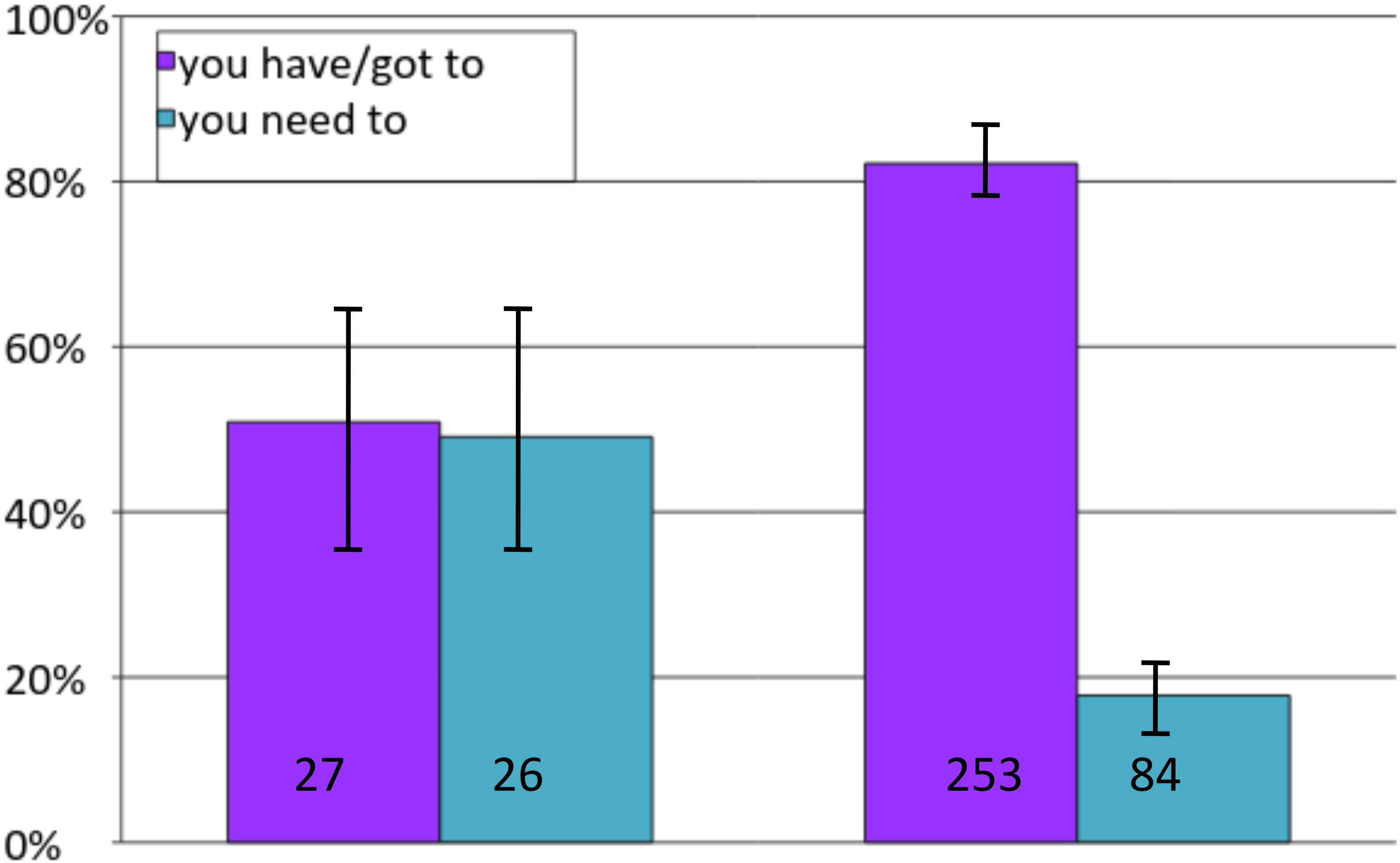
- Dwight craves authority, is a busybody, thinks he knows what is best for the office; is not socially sensitive

- Prediction: Dwight will use a higher rate of *you need to* in comparison to *have to/got to* than others

- *Need to* from someone *without* legitimacy can misfire – potentially partly explaining why Dwight is perceived as irritating

Work and some slides from [Lelia Glass, NWAV 2014]

Prediction: Dwight will use a higher proportion of *you need to* because he craves authority, thinks he knows what is good for the office, not socially sensitive



Work and some slides from [Lelia Glass, NWAV 2014]

A phonological corpus study: names

- Hypothesis: Prosody (phonological rhythm) influences choice of baby names
- Corpus: Facebook names data 100m+ names
- First observations:
    - Names alliterate (i.e. Peter Potts > Rodger Potts) more than would happen by chance.
    - Names avoid adjacent stresses (i.e. Súsan Smíth > Suzánne Smíth) more than would happen by chance.

Work by Stephanie Shih and Tyler Schnoebelen

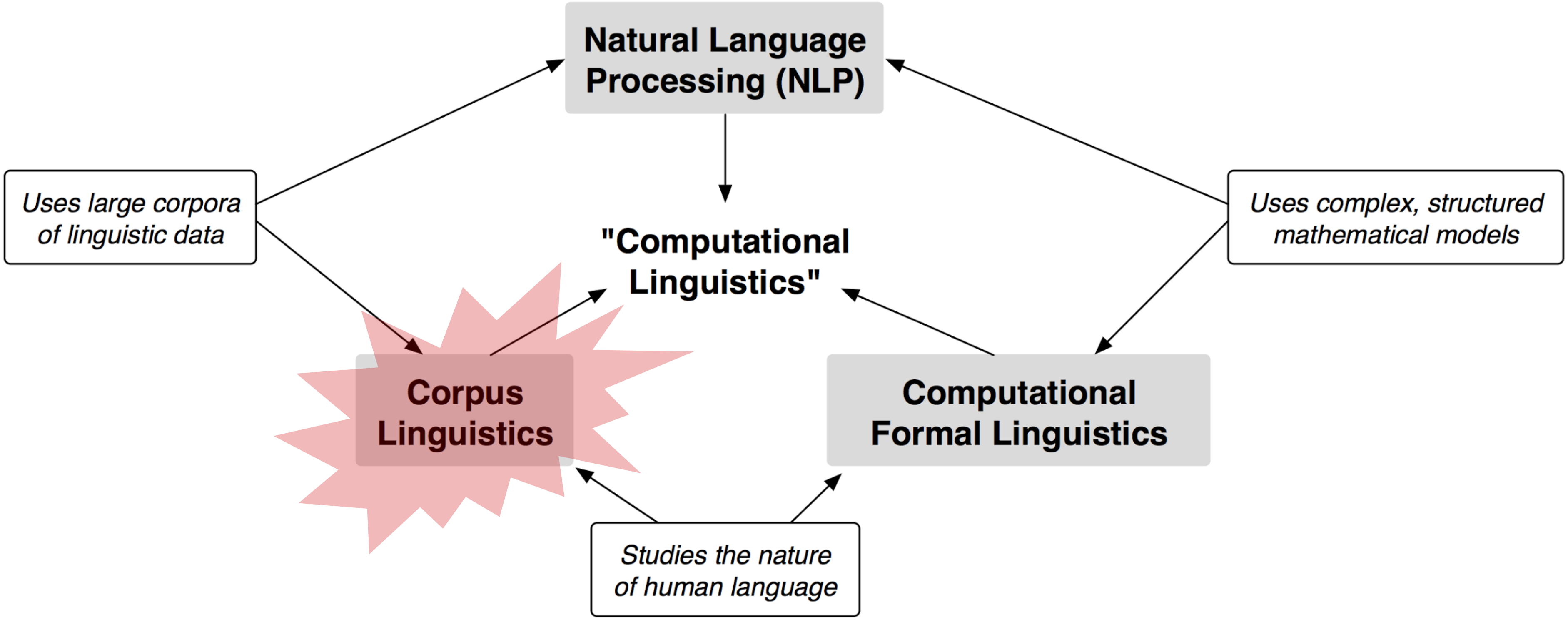
A phonological corpus study: names

- If people were able to choose full names, without the constraint of using family surnames, we would expect these effects to be stronger.

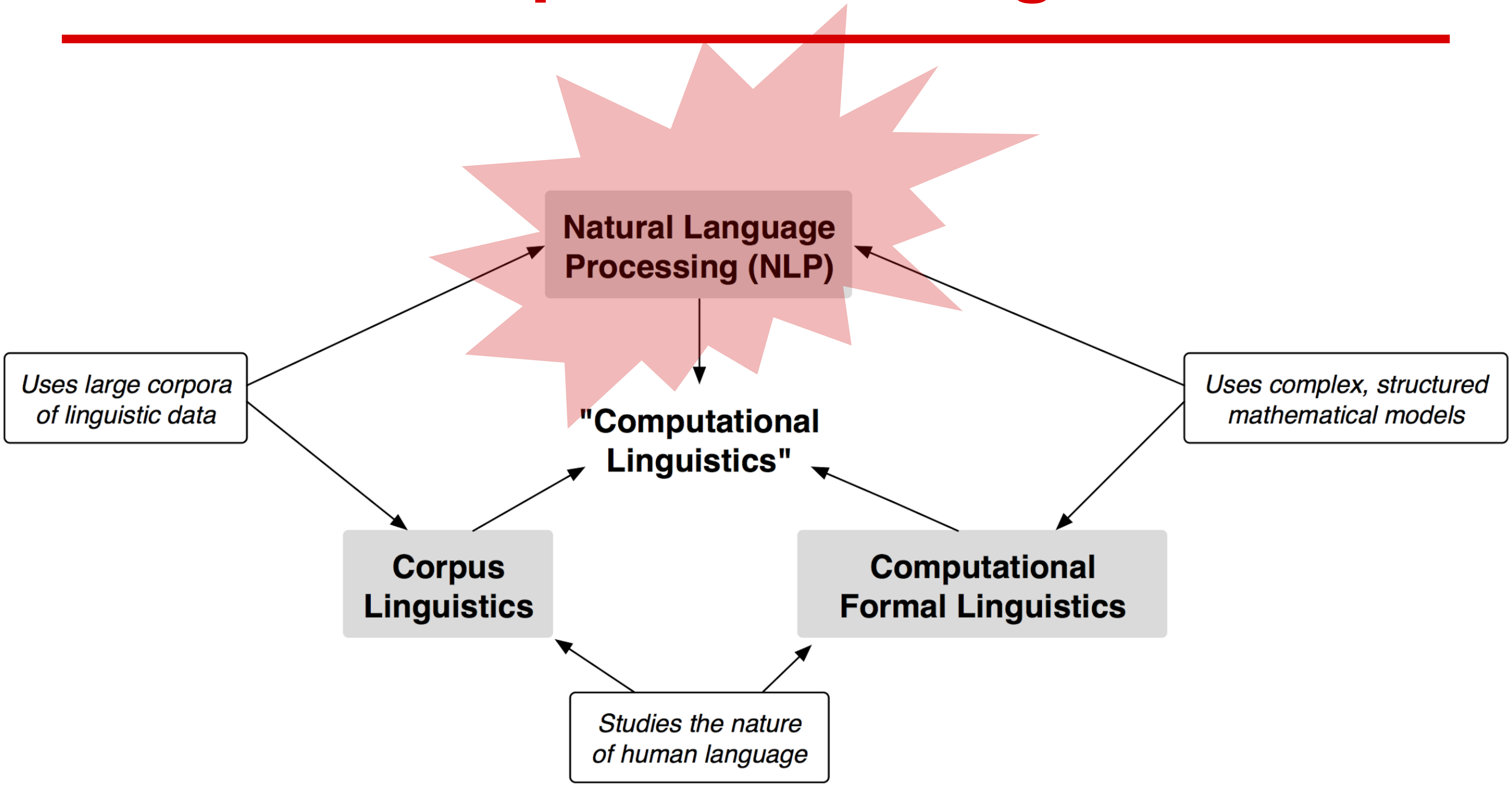
  ⇒ Study porn star names!

  (work in progress)

What is computational linguistics?



What is computational linguistics?



Natural language processing

The branch of artificial intelligence that attempts to produce computational tools which can use language.

- Even partial successes can be very useful:
  - Machine translation (Google Translate, ...)
  - Question answering (Siri, Wolfram Alpha, ...)
  - Spelling and grammar correction
  - Information extraction (Gmail calendar events)
  - Sentiment analysis (predicting stocks with Twitter)
  - Spam filtering
  - Speech recognition

  ...

The history of NLP in one slide

- **Before 1990**:
  - Handwritten grammars

```
bank => banco
I went to [NP] => Fui a [NP]
```

  - Handwritten dictionaries
  - Carefully designed complex models
- **Early 1990s**: The statistical revolution!
  - Statistical models + machine learning

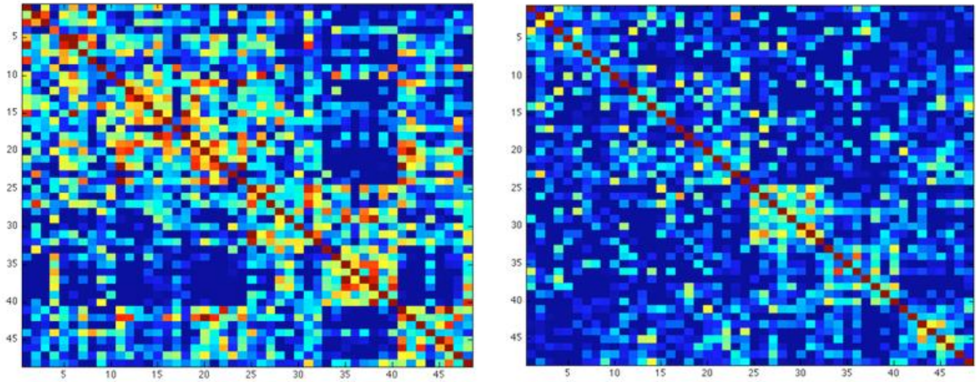
```
I went to the bank. There are reeds along the bank...
Fui al banco. Hay cañas a lo largo de la orilla…
  bank => {banco (.8), orilla (.18), cajero automático
           (0.0012), rana (0.001)...}
```

- **Early 2010s (ongoing)**: The deep learning revolution!

Deep learning (deep neural nets)

```
Training data:
{I went to the bank.,
 Fui al banco.}
{There are reeds along the bank.,
 Hay cañas a lo largo de la orilla.}
...
```

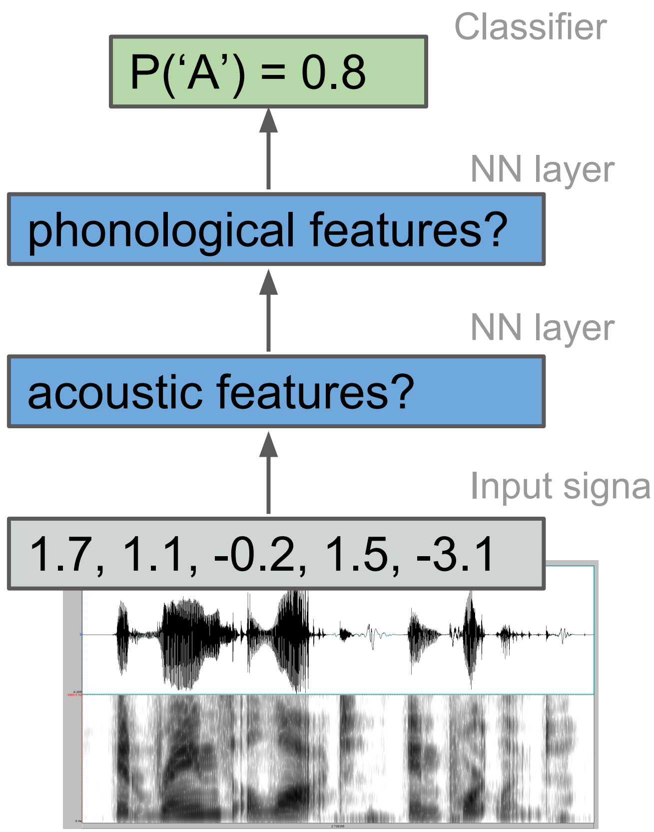
The other day when I was at the **bank** I ... =>



=> ... **banco** ...

Deep neural networks

- No neurons were harmed in the making of this network…
- Each NN layer implements a mathematically simple function from real vectors to real vectors.
- Layer functions parameters can be learned from (labeled) examples.

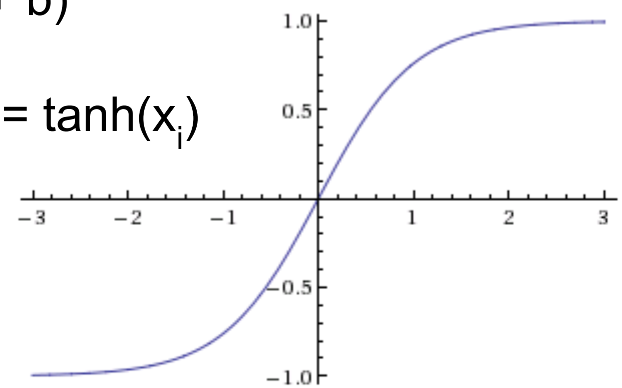


(Feedforward) deep neural networks

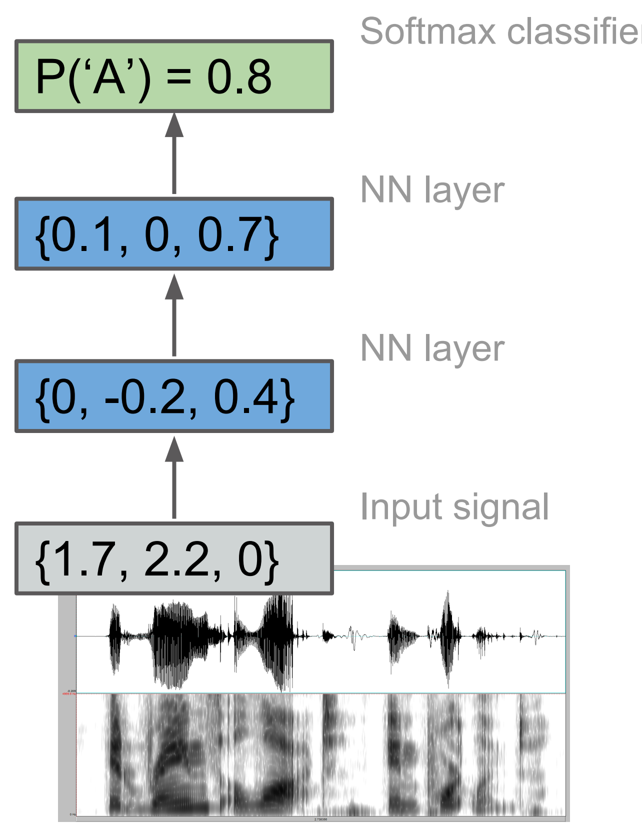
- Defines a function from a real-valued vector input to (usually) a distribution over labels
- Each NN layer implements a simple parametric function, often:

  $$y = f(Mx + b)$$

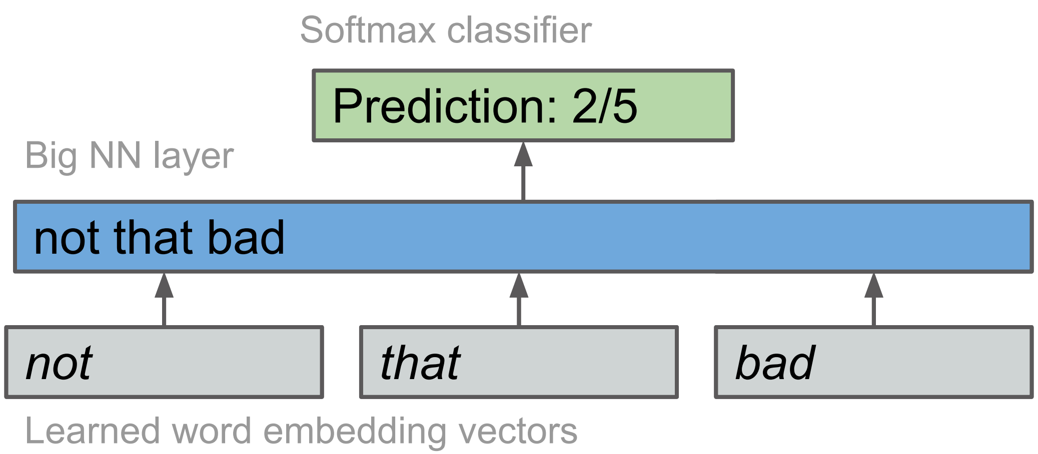
  $$f(x_i) = \tanh(x_i)$$



- Parameters can be learned from (labeled) examples:
  - Backpropagation to compute gradients
  - Learn using stochastic gradient descent

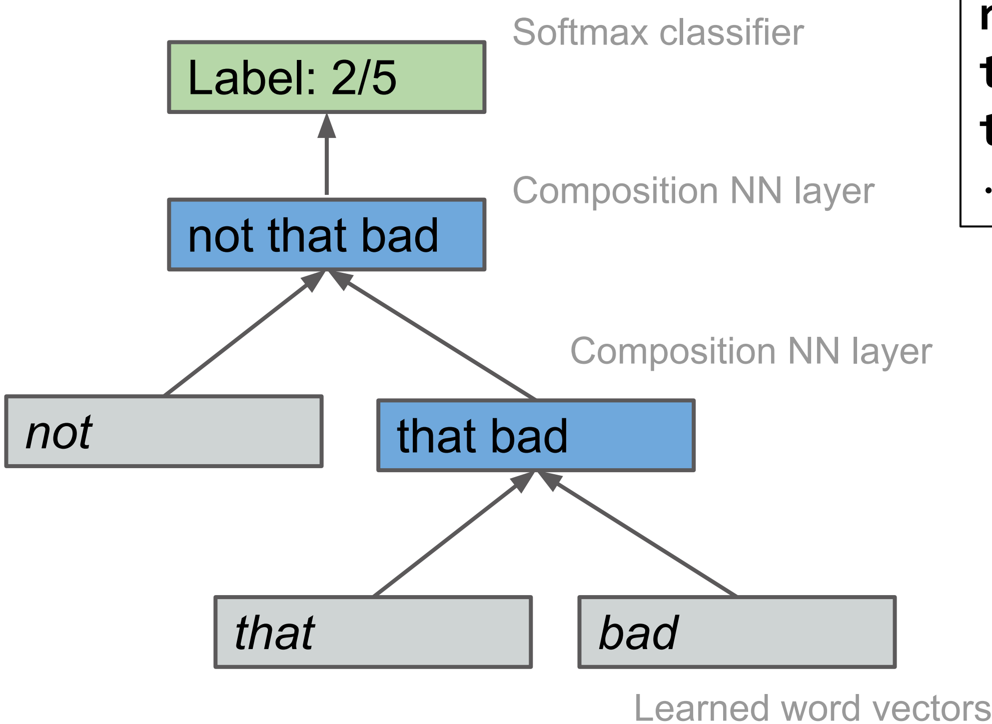


Neural networks for text



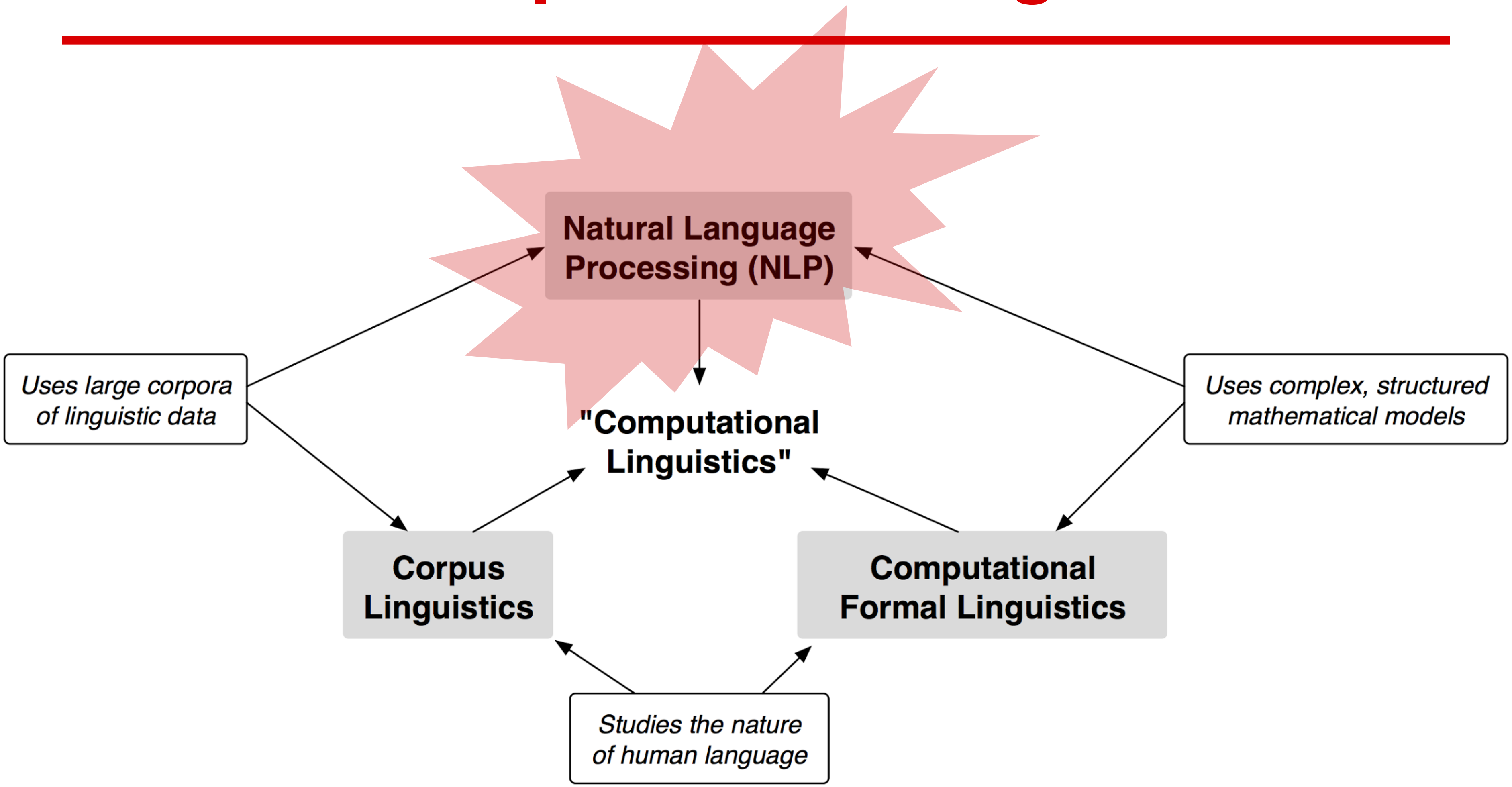

```
WORD EMBEDDINGS:
a = {-0.2, -9.3, 1.1}
bad = {0.0, 0.4, 19.9}
not = {0.1, 3.3, 2.0}
terrible = {0.1, 0.4, 17.9}
that = {0.1, 3.9, -0.1}
...
```

(Tree) neural networks for text

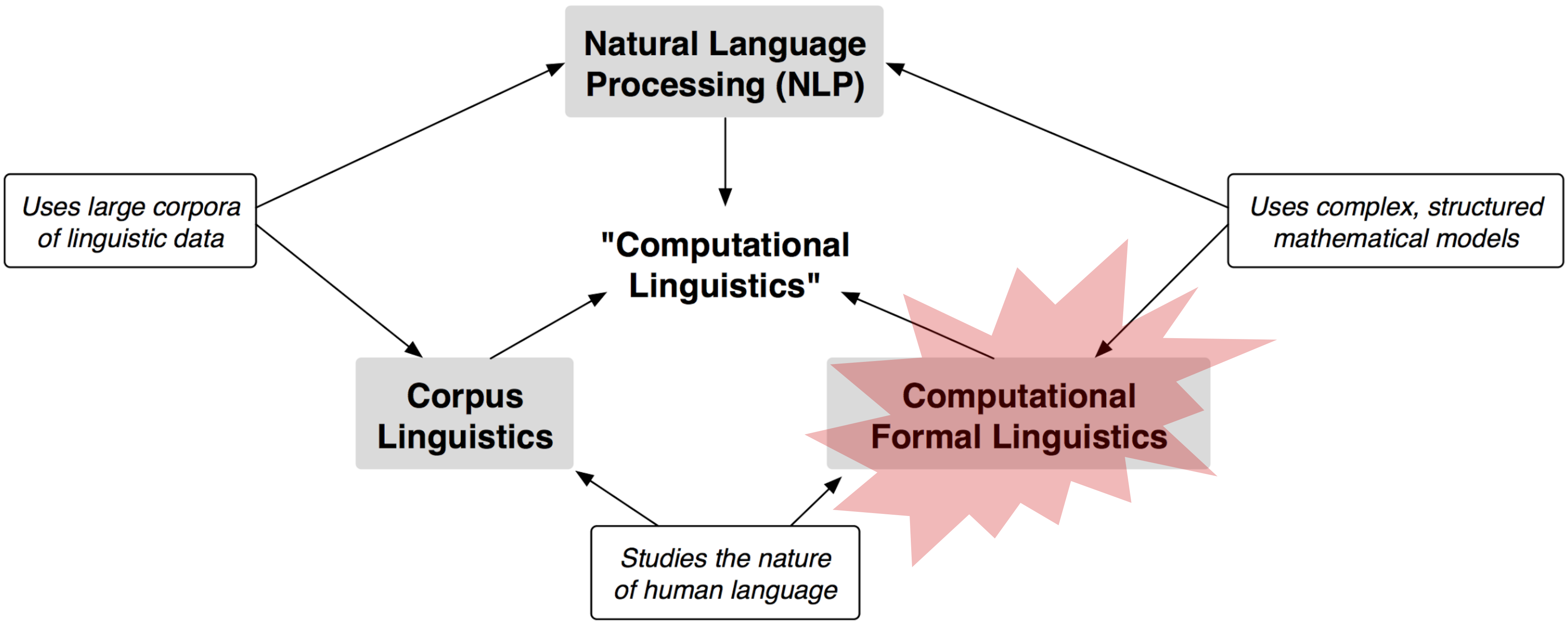



```
WORD EMBEDDINGS:
a = {-0.2, -9.3, 1.1}
bad = {0.0, 0.4, 19.9}
not = {0.1, 3.3, 2.0}
terrible = {0.1, 0.4, 17.9}
that = {0.1, 3.9, -0.1}
...
```

What is computational linguistics?



What is computational linguistics?



Computational formal linguistics

Using computational tools to better understand formal theories.

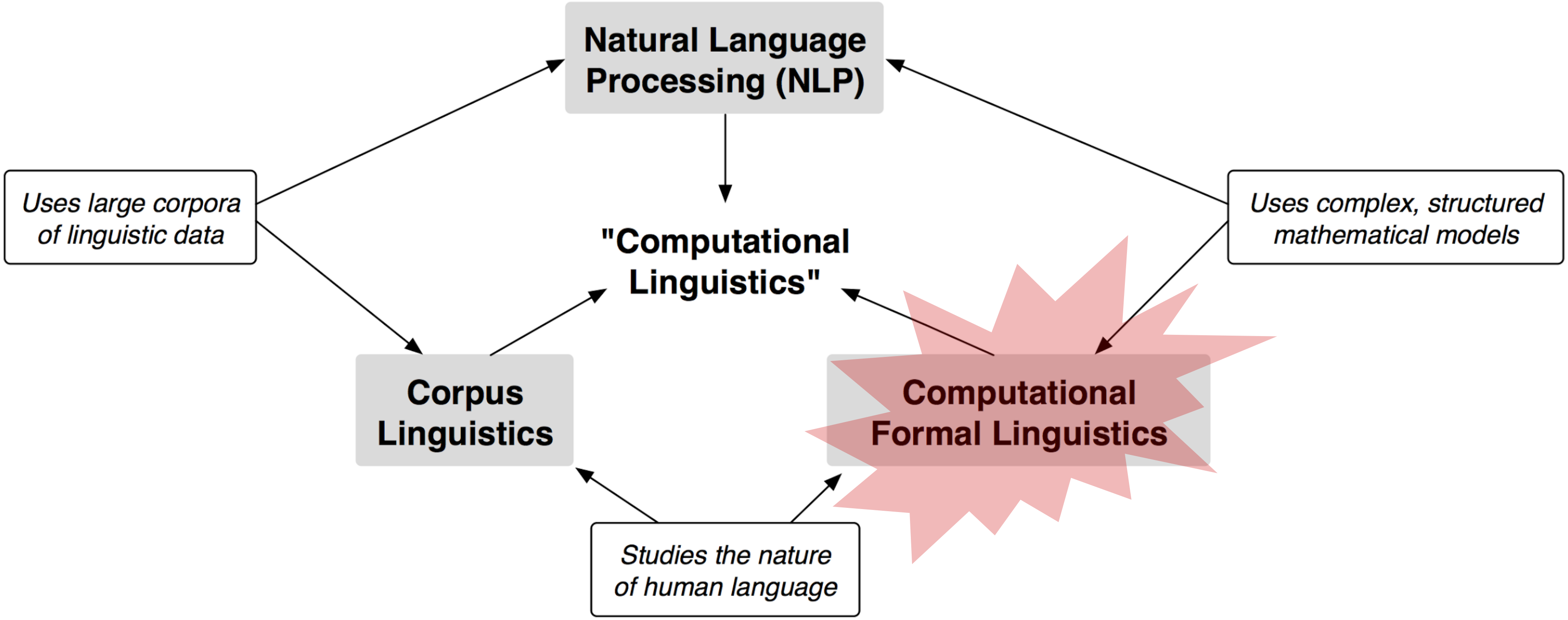
What I cannot create, I do not understand.

(Feynmann, 1988)

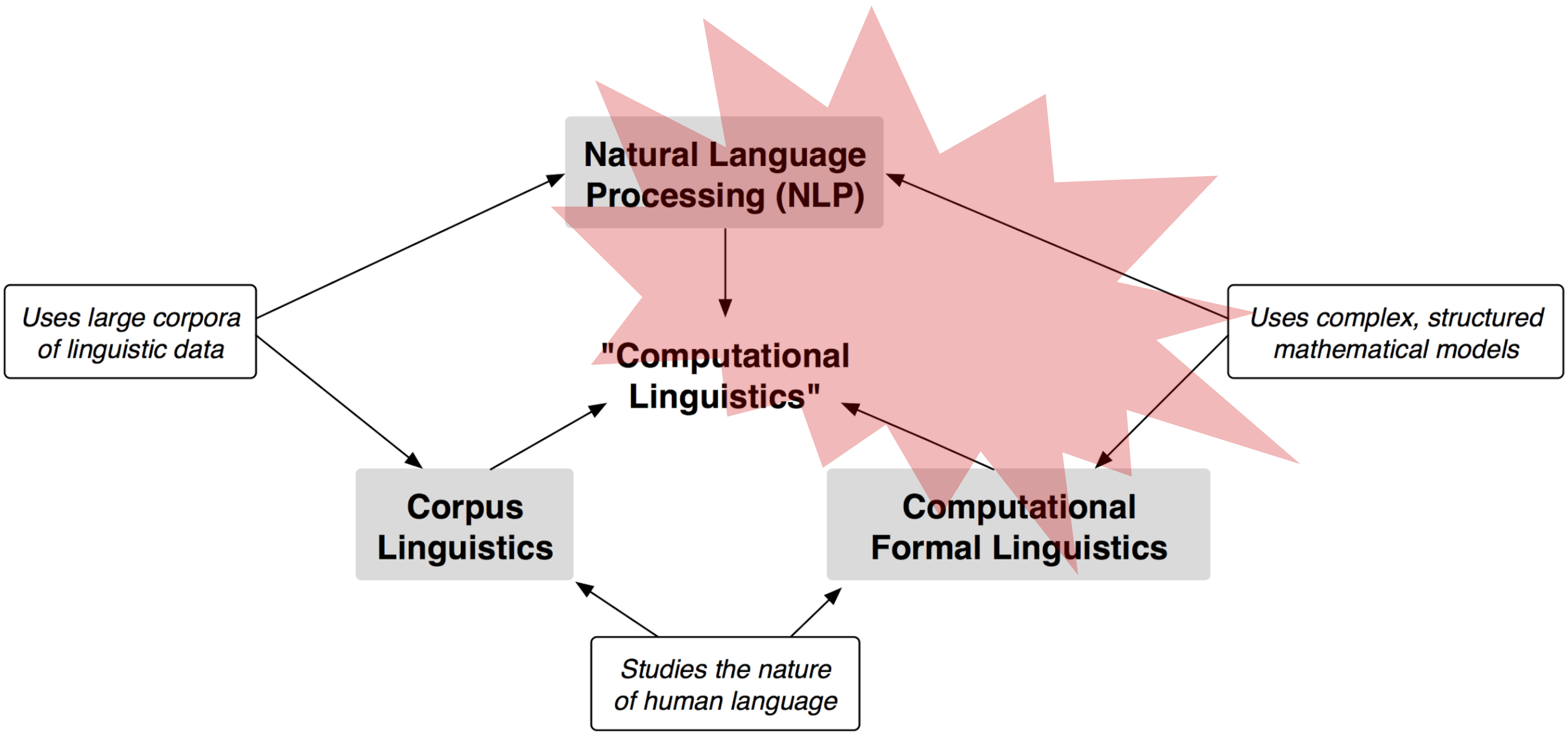
Examples of research in CFL

- If you have a theory about how languages change over time…
  - Write a program to reconstruct historical languages and see if they match what we know.
- If you have a formal theory of how humans transform an acoustic signal into a string of phonemes…
  - ...implement it and see if it makes similar mistakes to humans.
- If you have a syntax of a language that's meant to explain what types of sentence are grammatical…
  - ...implement it and see if it can distinguish between sentences from a real corpus and invented fake examples.

What is computational linguistics?



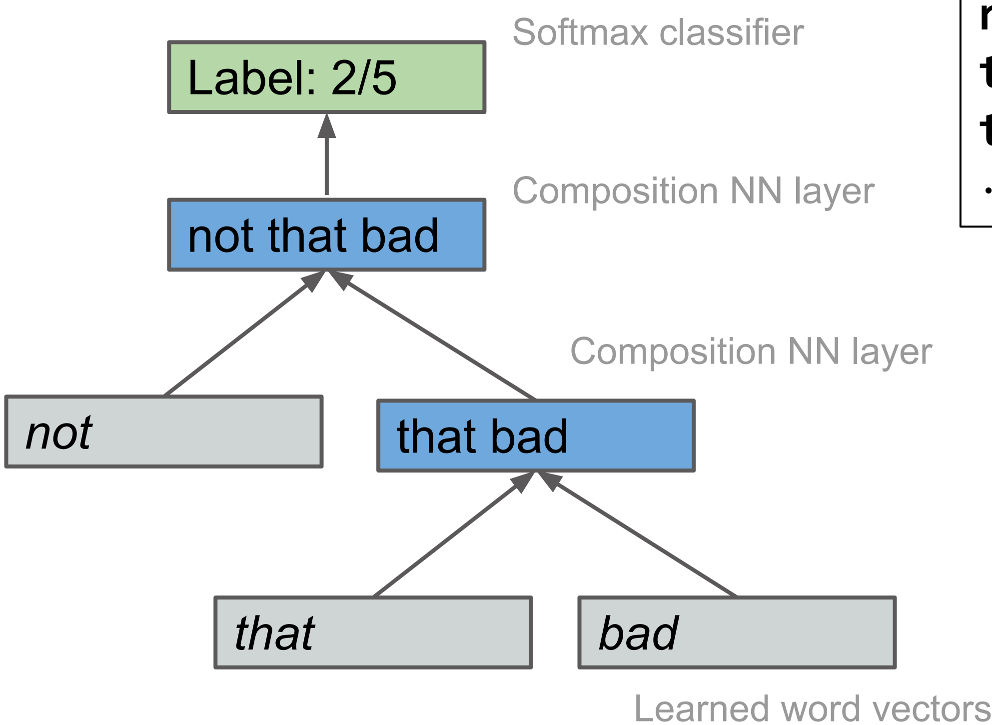
What is computational linguistics?



The big question

How well are supervised neural network models able to learn representations of sentence meaning?

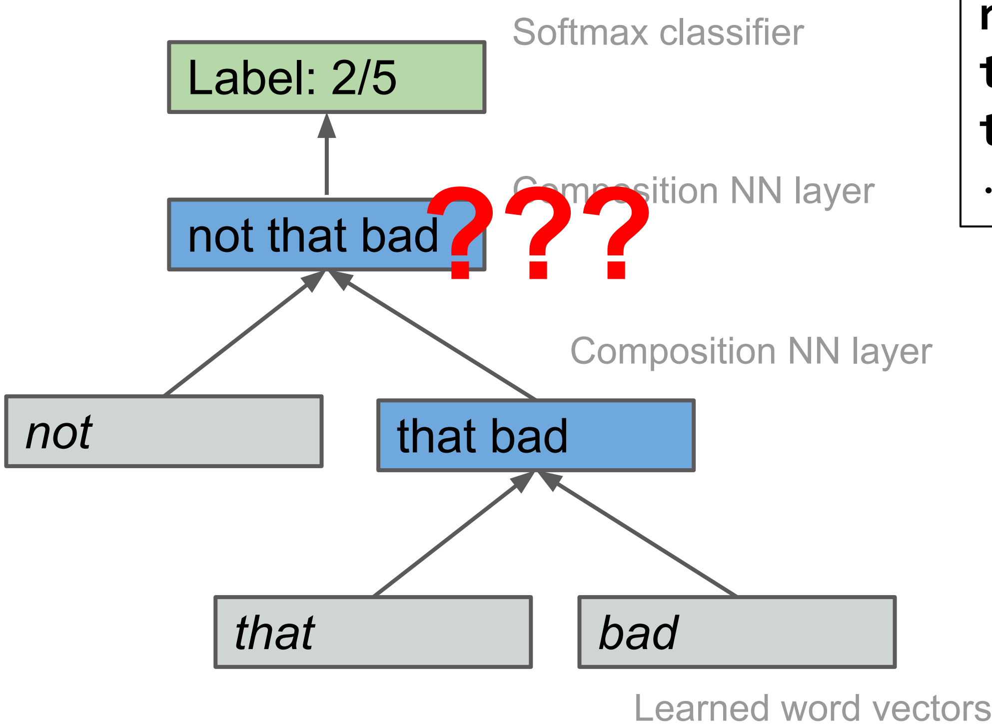
(Tree) neural networks for text




```
WORD EMBEDDINGS:
a = {-0.2, -9.3, 1.1}
bad = {0.0, 0.4, 19.9}
not = {0.1, 3.3, 2.0}
terrible = {0.1, 0.4, 17.9}
that = {0.1, 3.9, -0.1}
...
```

Socher et al. 2011

(Tree) neural networks for text




```
WORD EMBEDDINGS:
a = {-0.2, -9.3, 1.1}
bad = {0.0, 0.4, 19.9}
not = {0.1, 3.3, 2.0}
terrible = {0.1, 0.4, 17.9}
that = {0.1, 3.9, -0.1}
...
```

Socher et al. 2011

The big question

How well are supervised neural network models able to learn representations of sentence meaning?

The big question

How well are supervised neural network models able to learn representations of sentence meaning?

The big question

How well are supervised neural network models able to learn representations of sentence meaning?

Don't ask what meanings are. Ask what they do, and find something that does that.

-David Lewis, paraphrased

The task: Natural language inference

A: James Byron Dean refused to move without blue jeans

B: James Dean didn’t dance without pants

{**entailment**, contradiction, neither}

The task: Natural language inference

Claim: Simple task to define, but engages the full complexity of compositional semantics:

- Lexical entailment
- Quantification
- Coreference
- Lexical/scope ambiguity
- Commonsense knowledge
- Propositional attitudes
- Modality
- Factivity and implicativity

...

Experiments

Experimental approach: Train on relational statements generated from some formal system, test on other such relational statements.

Experiments

Experimental approach: Train on relational statements generated from some formal system, test on other such relational statements.

The model needs to:

- Learn the relations between individual words.

Formulating a learning problem

Training data:

dance	***entails***	*move*
waltz	***neutral***	*tango*
tango	***entails***	*dance*
sleep	***contradicts***	*dance*
waltz	***entails***	*dance*

Memorization (training set):

dance ***???*** *move*

waltz ***???*** *tango*

Generalization (test set):

sleep ***???*** *waltz*

tango ***???*** *move*

MacCartney's natural logic

An implementable logic for natural language inference without logical forms. (MacCartney and Manning '09)

- Sound logical interpretation (Icard and Moss '13)

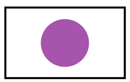
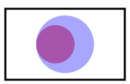
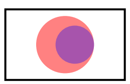
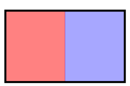
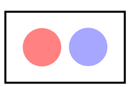
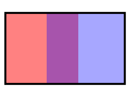
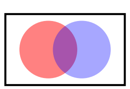
P	*James Dean*	*refused to*			*move*	*without*	*blue*	*jeans*
H	*James Byron Dean*		*did*	*n't*	*dance*	*without*		*pants*
edit index	1	2	3	4	5	6	7	8
edit type	SUB	DEL	INS	INS	SUB	MAT	DEL	SUB
lex feats	strsim= 0.67	implic: –/o	cat:aux	cat:neg	hypo			hyper
lex entrel	$=$	$\mid$	$=$	$\wedge$	$\sqsupset$	$=$	$\sqsubset$	$\sqsubset$
projectivity	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\downarrow$	$\downarrow$	$\uparrow$	$\uparrow$
atomic entrel	$=$	$\mid$	$=$	$\wedge$	$\sqsubset$	$=$	$\sqsubset$	$\sqsubset$

inversion

Natural logic: relations

Seven possible relations between phrases/sentences:

Slide from Bill MacCartney

	$x \equiv y$	equivalence	*couch* ≡ *sofa*
	$x \sqsubset y$	forward entailment (strict)	*crow* ⊏ *bird*
	$x \sqsupset y$	reverse entailment (strict)	*European* ⊐ *French*
	$x \wedge y$	negation (exhaustive exclusion)	*human* ^ *nonhuman*
	$x \mid y$	alternation (non-exhaustive exclusion)	*cat* | *dog*
	$x \smile y$	cover (exhaustive non-exclusion)	*animal* ‿ *nonhuman*
	$x \# y$	independence	*hungry* # *hippo*

Natural logic: relation joins

	$\equiv$	$\sqsubset$	$\sqsupset$	$\wedge$	$\mid$	$\smile$	$\#$
$\equiv$	$\equiv$	$\sqsubset$	$\sqsupset$	$\wedge$	$\mid$	$\smile$	$\#$
$\sqsubset$	$\sqsubset$	$\sqsubset$	$\cdot$	$\mid$	$\mid$	$\cdot$	$\cdot$
$\sqsupset$	$\sqsupset$	$\cdot$	$\sqsupset$	$\smile$	$\cdot$	$\smile$	$\cdot$
$\wedge$	$\wedge$	$\smile$	$\mid$	$\equiv$	$\sqsupset$	$\sqsubset$	$\#$
$\mid$	$\mid$	$\cdot$	$\mid$	$\sqsubset$	$\cdot$	$\sqsubset$	$\cdot$
$\smile$	$\smile$	$\smile$	$\cdot$	$\sqsupset$	$\sqsupset$	$\cdot$	$\cdot$
$\#$	$\#$	$\cdot$	$\cdot$	$\#$	$\cdot$	$\cdot$	$\cdot$

MacCartney's join table: $a\ R\ b \wedge b\ R'\ c \vdash a\ \{R \bowtie R'\}\ c$

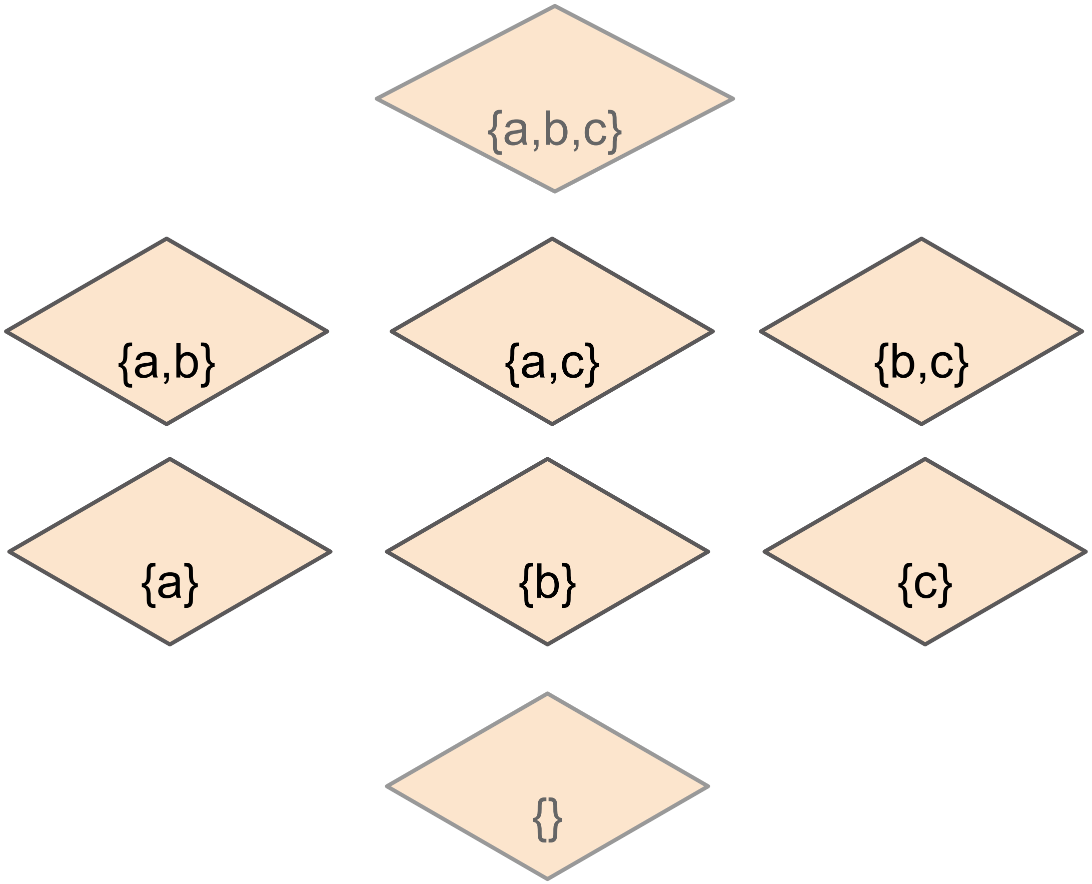
{animal $\sqsupset$ cat, cat $\sqsupset$ kitten} $\vdash$ animal $\sqsupset$ kitten
{cat $\sqsubset$ animal, animal ^ non-animal} $\vdash$ cat | non-animal

Natural logic: relation joins

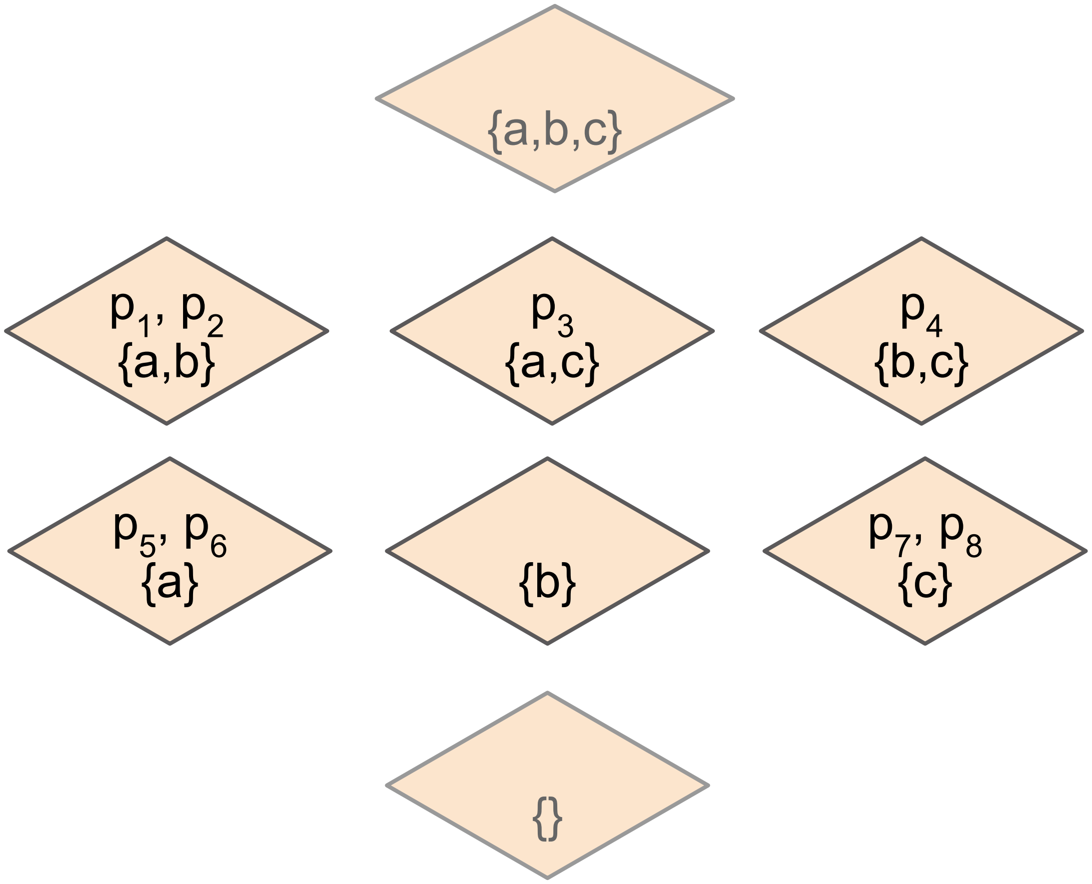
	$\equiv$	$\sqsubset$	$\sqsupset$	$\wedge$	$\mid$	$\smile$	$\#$
$\equiv$	$\equiv$	$\sqsubset$	$\sqsupset$	$\wedge$	$\mid$	$\smile$	$\#$
$\sqsubset$	$\sqsubset$	$\sqsubset$	$\cdot$	$\mid$	$\mid$	$\cdot$	$\cdot$
$\sqsupset$	$\sqsupset$	$\cdot$	$\sqsupset$	$\smile$	$\cdot$	$\smile$	$\cdot$
$\wedge$	$\wedge$	$\smile$	$\mid$	$\equiv$	$\sqsupset$	$\sqsubset$	$\#$
$\mid$	$\mid$	$\cdot$	$\mid$	$\sqsubset$	$\cdot$	$\sqsubset$	$\cdot$
$\smile$	$\smile$	$\smile$	$\cdot$	$\sqsupset$	$\sqsupset$	$\cdot$	$\cdot$
$\#$	$\#$	$\cdot$	$\cdot$	$\#$	$\cdot$	$\cdot$	$\cdot$

Can our NNs learn to make these inferences over pairs of embedding vectors?

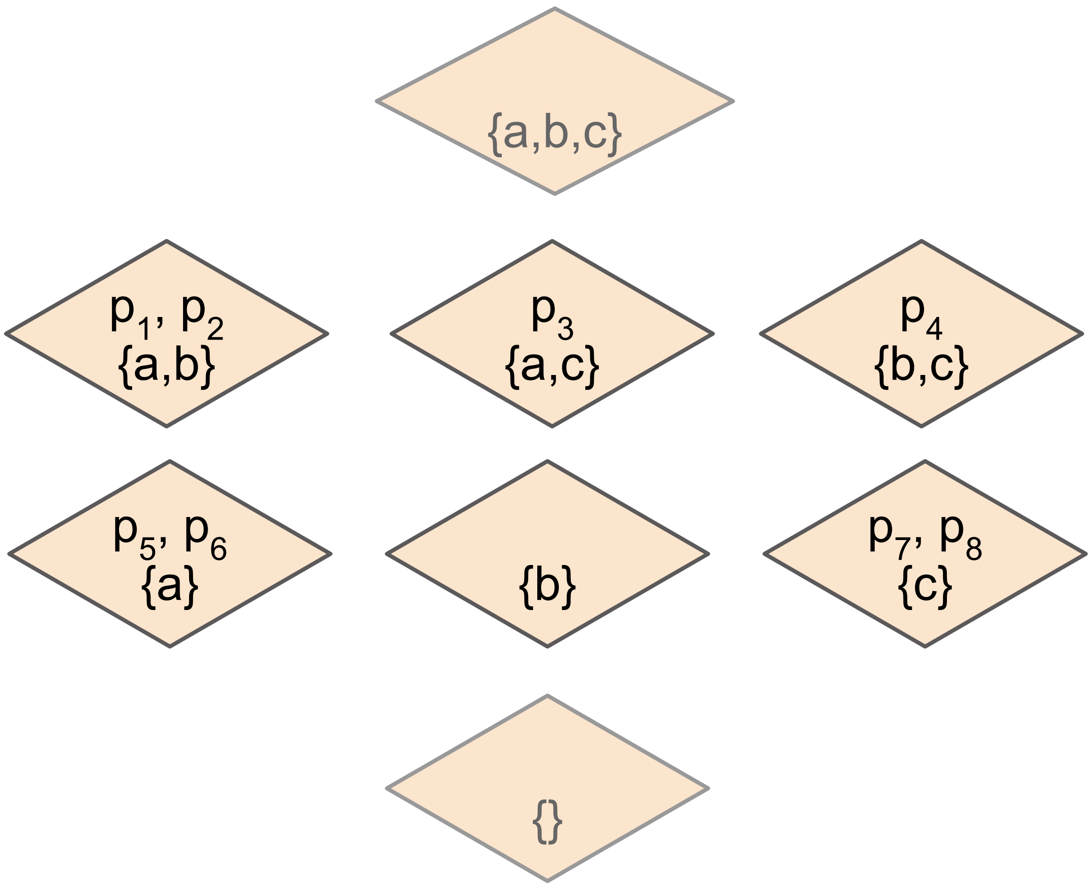
Lexical relations: data generation



Lexical relations: data generation



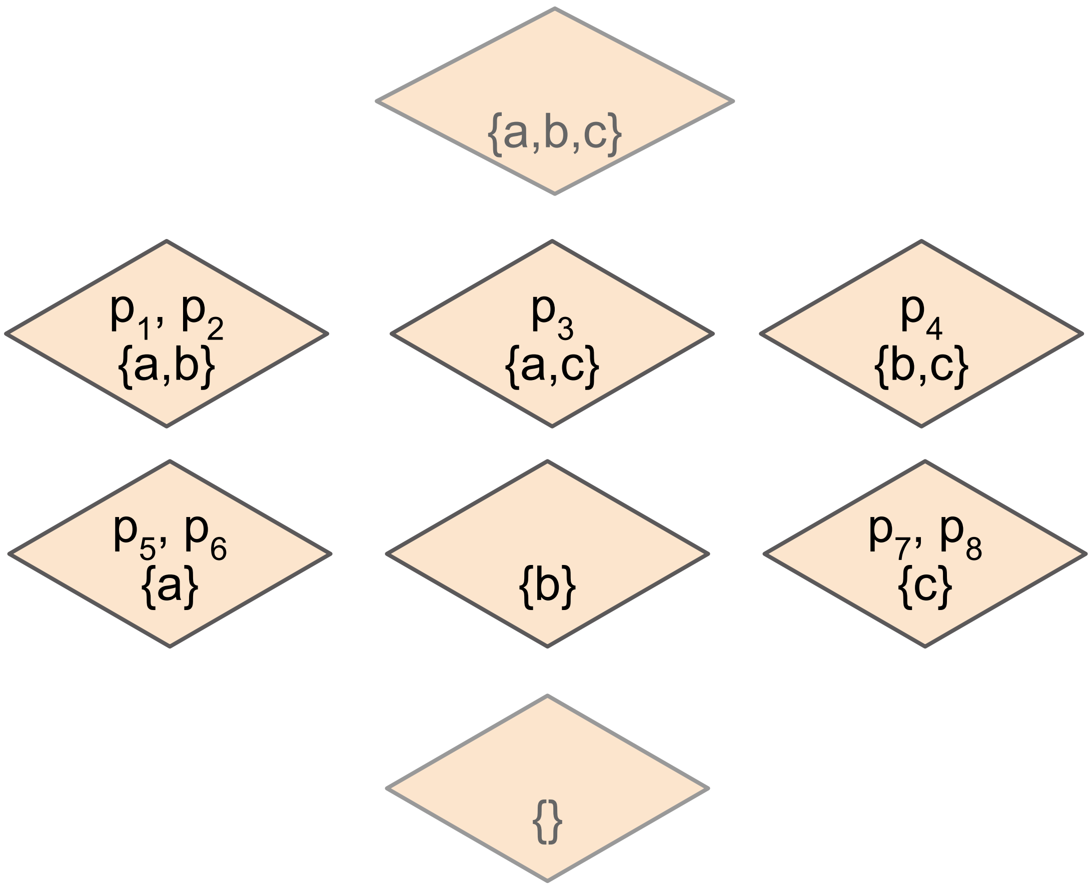
Lexical relations: data generation



Extracted relations:

$p_1 \equiv p_2$
$p_1 \wedge p_7$
$p_1 \sqsupset p_5$
$p_4 \sqsupset p_8$
$p_2 \sqsupset p_5$
$p_5 \equiv p_6$
$p_5 \mid p_7$
$p_7 \sqsubset p_4$
$p_7 \wedge p_1$
$p_8 \sqsupset p_1$

Lexical relations: data generation



Train:

$p_1 \equiv p_2$

$p_1 \sqsupset p_5$

$p_4 \sqsupset p_8$

$p_5 \mid p_7$

$p_7 \wedge p_1$

Test:

$p_1 \wedge p_7$

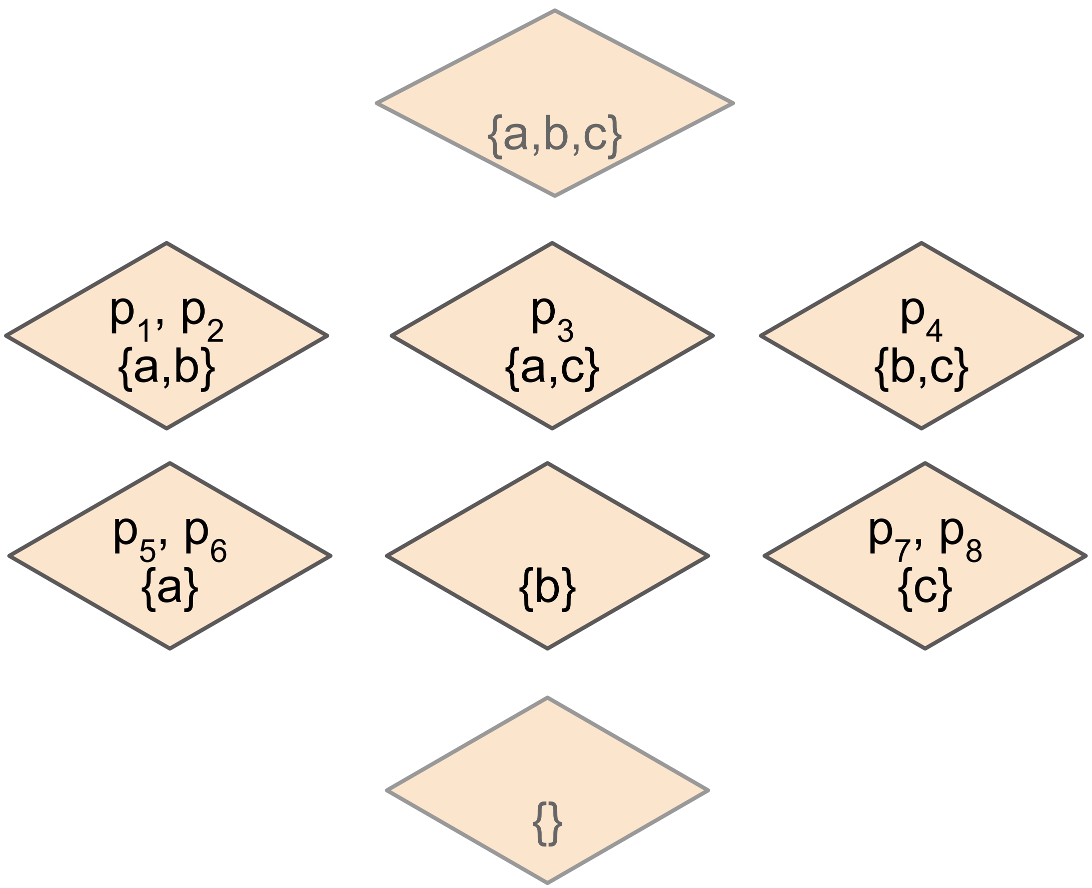
$p_2 \sqsupset p_5$

$p_5 \equiv p_6$

$p_7 \sqsubset p_4$

$p_8 \sqsupset p_4$

Lexical relations: data generation



Train:

$p_1 \equiv p_2$

$p_1 \sqsupset p_5$

$p_4 \sqsupset p_8$

$p_5 \mid p_7$

$p_7 \wedge p_1$

Test:

$p_1 \wedge p_7$

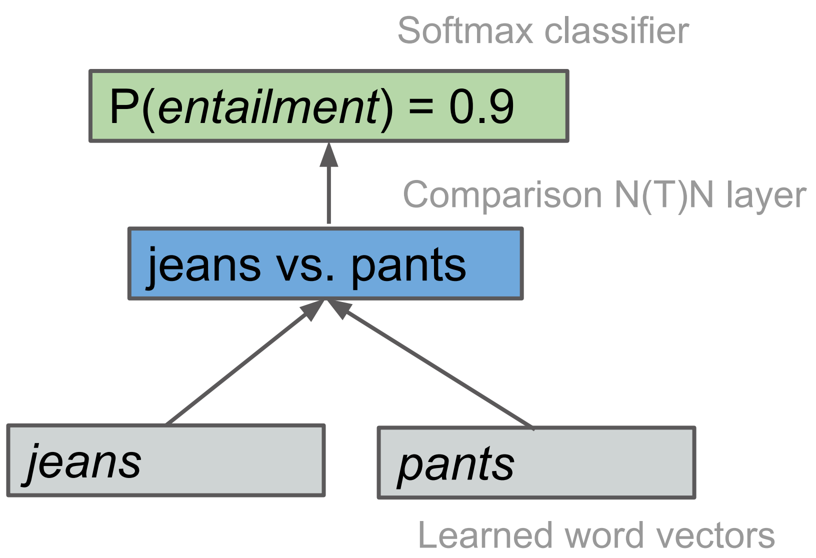
$p_2 \sqsupset p_5$

$p_5 \equiv p_6$

$p_7 \sqsubset p_4$

$p_8 \sqsupset p_4$

A minimal NN for lexical relations



Lexical relations: training

- 80 random terms (p_1 - p_{80})

- 6400 statements, yielding:
  - 3200 training examples
  - about 2900 provable test examples
    (~7% not provable)

Lexical relations: results

	Train	Test
# only	53.8 (10.5)	53.8 (10.5)
15d NN	99.8 (99.0)	94.0 (87.0)
15d NTN	**100 (100)**	**99.6 (95.5)**

- Both models tuned, then trained to convergence on five randomly generated datasets
- Reported figures: % correct (macroaveraged F1)
- Both NNs used 15d embeddings, 75d comparison layer

Lexical relations: Conclusions

- Success! NTNs can learn lexical entailment networks
  - No special optimization techniques required
  - Good generalization even with small training sets

Experiments

Experimental approach: Train on relational statements generated from some formal system, test on other such relational statements.

The model needs to:

- Learn the relations between individual words.

Recursion in propositional logic

Experimental approach: Train on relational statements generated from some formal system, test on other such relational statements.

The model needs to:

- Learn the relations between individual words.
- Learn how to construct the relations between phrases from words.
  - This needs to use recursion!

$a \equiv a$, $a \wedge (\text{not } a)$, $a \equiv (\text{not } (\text{not } a))$, ...

Recursion in propositional logic

Data: randomly generated sentences with *and*, *or*, and *not*

- 6 proposition variables (a-f), at most 4 per example
- Propositions are variables over unknown truth values (2^{64} possible representations)
- Train on statements with at most 4 operators, test with more.

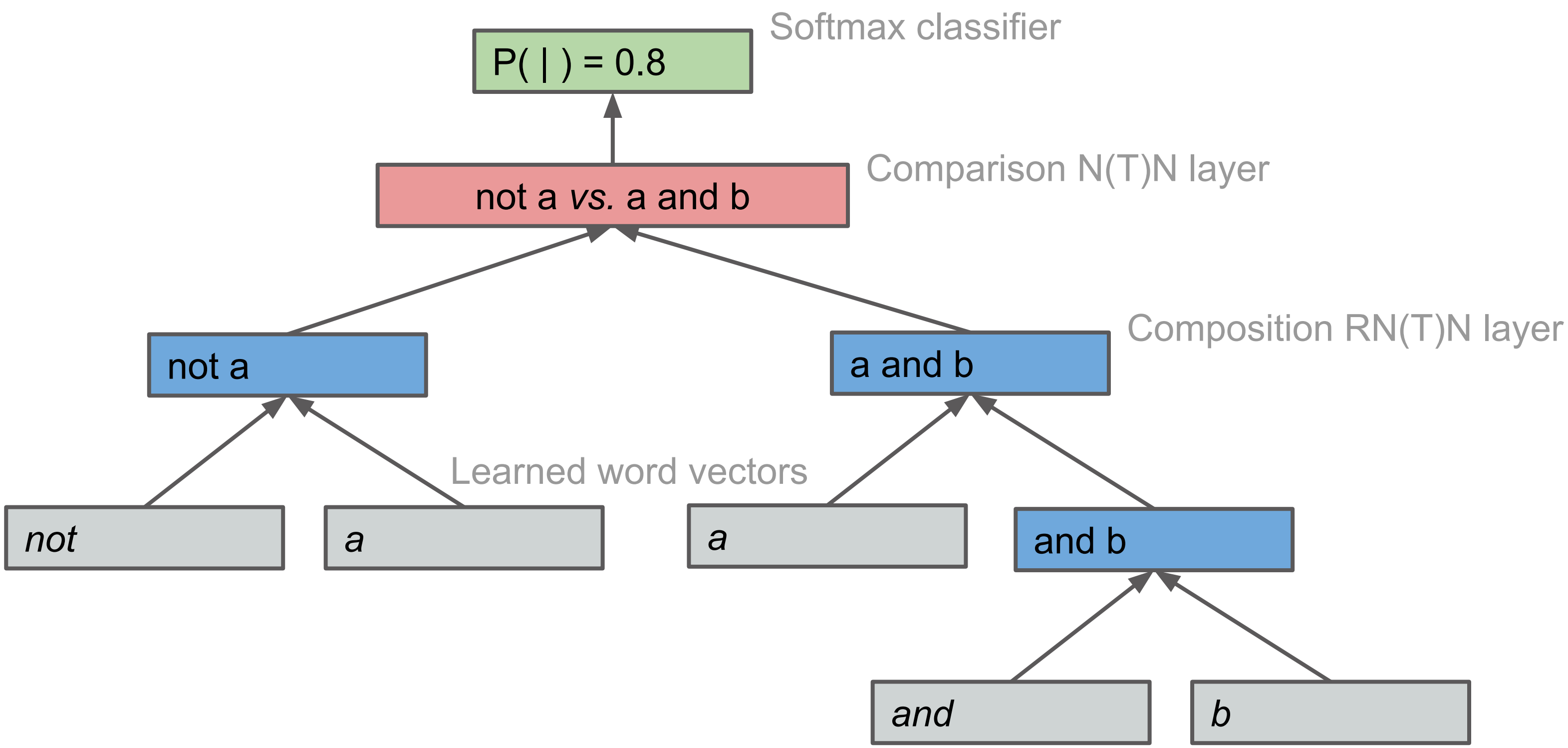
Formula	Interpretation
a, b, c, d, e, f	$[\![x]\!] \in \{\mathsf{T}, \mathsf{F}\}$
not φ	T iff $[\![\varphi]\!] = \mathsf{F}$
$(\varphi$ *and* $\psi)$	T iff $\mathsf{F} \notin \{[\![\varphi]\!], [\![\psi]\!]\}$
$(\varphi$ *or* $\psi)$	T iff $\mathsf{T} \in \{[\![\varphi]\!], [\![\psi]\!]\}$

(a) Well-formed formulae. φ and ψ range over all well-formed formulae, and $[\![\cdot]\!]$ is the interpretation function mapping formulae into $\{\mathsf{T}, \mathsf{F}\}$.

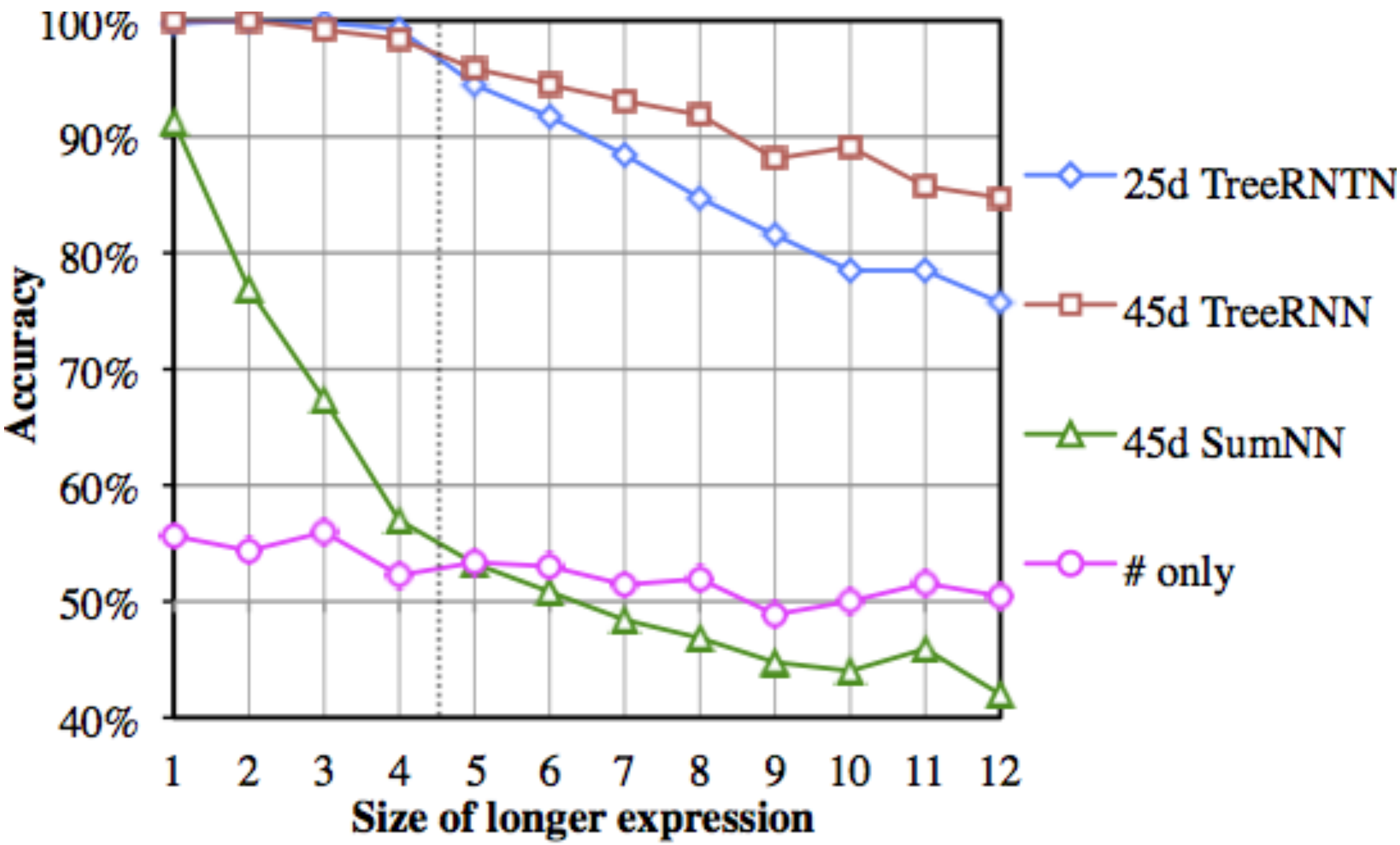
not a	$\wedge$	a
not not a	$\equiv$	a
a	$\sqsubset$	$(a$ *or* $b)$
a	$\sqsupset$	$(a$ *and* $b)$
not$($*not* a *and not* $b)$	$\equiv$	$(a$ *or* $b)$

(b) Examples of statements about relations between well-formed formulae, defined in terms of sets of satisfying interpretation functions $[\![\cdot]\!]$.

NLI with Tree NNs



Recursion in propositional logic



Experiments

Experimental approach: Train on relational statements generated from some formal system, test on other such relational statements.

The model needs to:

- Learn the relations between individual words.
- Learn how to construct the relations between phrases from words.

Quantifiers

Experimental approach: Train on relational statements generated from some formal system, test on other such relational statements.

The model needs to:

- Learn the relations between individual words.
- Learn how to construct the relations between phrases from words.
- Quantifiers present some of the harder cases of both of these.

Two experiments

(most warthogs) walk	$\wedge$	(not-most warthogs) walk
(most mammals) move	$\#$	(not-most (not turtles)) move
(most (not pets)) (not swim)	$\sqsupset$	(not-most (not pets)) move
(no turtles) (not growl)	$\mid$	(no turtles) (not swim)
(no warthogs) swim	$\sqsupset$	(no warthogs) move
(no warthogs) move	$\sqsubset$	(no (not reptiles)) swim

Quantifier results

	Train	Test
Most freq. class (# only)	35.4%	35.4%
25d SumNN (sum of words)	96.9%	93.9%
25d TreeRNN	99.6%	99.2%
25d TreeRNTN	**100%**	**99.7%**

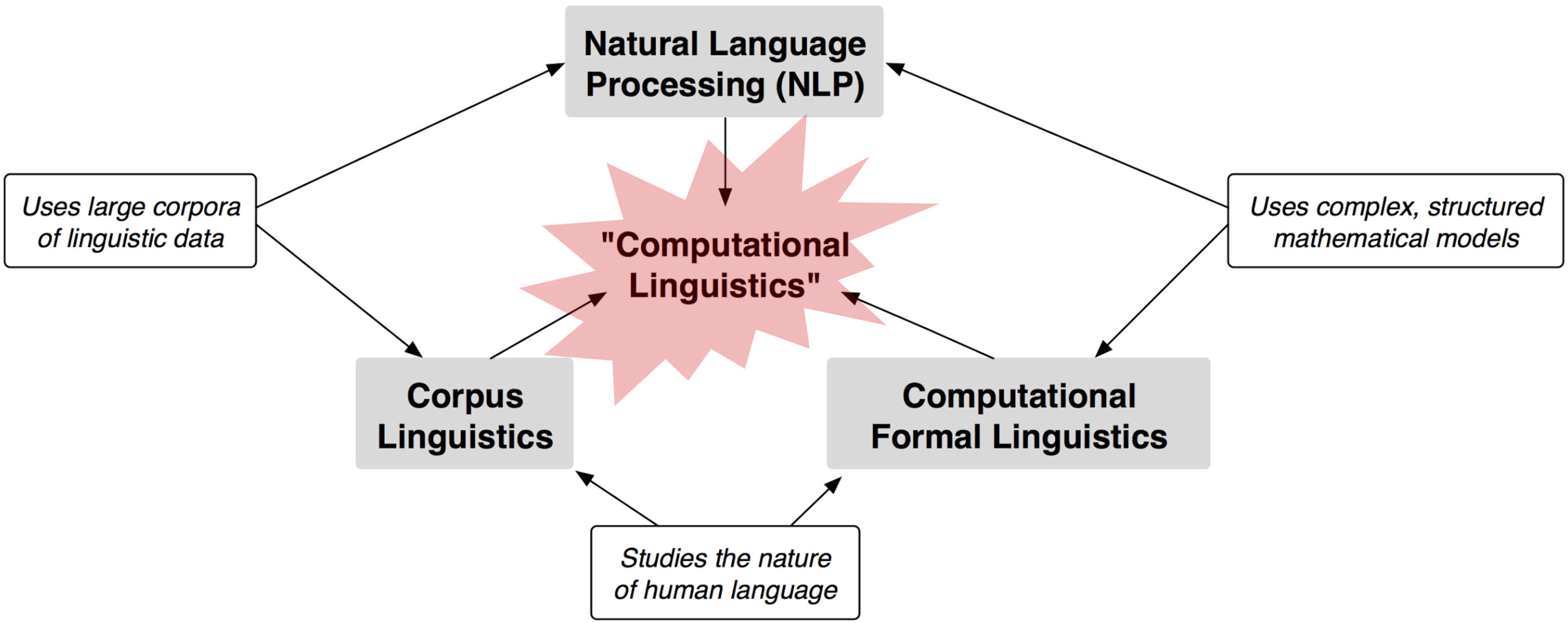
Summary: Artificial data

- Simple NNs can learn and reason about lexical relations.
- Tree structured models can learn recursive functions, and can apply them in structures that are (somewhat) larger than those seen in training.
- Tree structured models can learn to reason with quantifiers.

So what about real English?

More details in: Bowman, Potts, and Manning '15

What is computational linguistics?



Interested in computational formal ling?

- Take lots of linguistics classes, including graduate classes in ‘core’ areas like syntax and phonology.
- Learn to program, and take all formal/mathematical computer science classes that you can.

Interested in NLP?

- Learn to program
  - Java and Python are popular for NLP
- Learn some probability theory and machine learning.

  (experimental statistics, less so)
- Take classes in NLP
  - From Languages to Information
  - Natural Language Processing
  - Natural Language Understanding
  - Spoken Language Processing
  - Deep Learning for NLP

Interested in corpus linguistics?

- Take lots of linguistics classes!
  - Corpus classes are only offered occasionally, but a foundation in the linguistic questions is the hard/important part.
- Take some experimental statistics!
- Learn basic Unix and programming skills.

Questions about the talk? sbowman@stanford.edu