Recursive neural networks can learn logical semantics.

Samuel R. Bowman, Christopher Potts, and Christopher D. Manning

Natural language inference

- Tree structured (recursive) NNs: Designed to compute vector representations for sentence meaning by semantic composition
- Can they really do this? Limited evidence or theory so far for robust functional composition
- We test this ability on artificial and natural data.
- Task: Natural language inference (aka textual entailment)

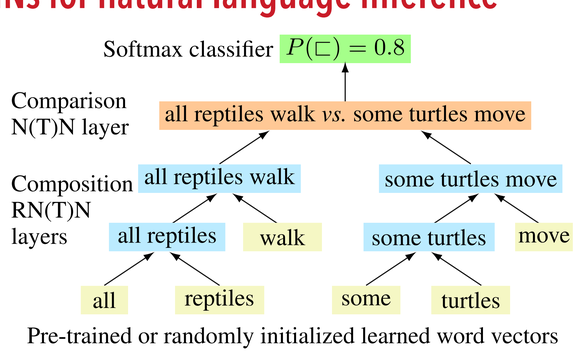
  *James Byron Dean refused to **move** without blue jeans*

  {**entails**, contradicts, neither}

  *James Dean didn't **dance** without pants*

- Simple to define/model as a classification task, but tests handling of every aspect of language meaning but grounding.

TreeRNNs for natural language inference



- Recursive layer fn.: sum, plain NN, or NTN (Chen et al. '13)

  $y_{NN} = f(M[x_l; x_r] + b)$ child vectors x_l and x_r are concatenated

  $y_{NTN} = f(x_l \mathbf{T}^{[1...N]} x_r + M[x_l; x_r] + b)$ T is a learned DxDxD tensor

- Dimension D of vectors is tuned (NB: the NTN layer has dramatically more parameters – $O(D^3)$ instead of $O(D^2)$)

Simulating logical composition with Natural Logic

- Formal logic for predicting natural language inference judgments (we use the formalism from MacCartney 2009)
- Defined over seven relations (see upper right) – exactly one applies for any two words/phrases/sentences of the same type
- We generate sentences from three artificial languages, then use implemented natural logic to label pairs of them.

Learning relation composition

- You'll never observe all word pairs – lexical relation composition fills in inevitable gaps in lexical knowledge for inference:

  if *{animal ⊐ cat, cat ⊐ kitten}* **then** *animal ⊐ kitten*

  if *{cat ⊏ animal, animal ^ non-animal}* **then** *cat | non-animal*

- We use artificial data: ~3k train pairs, 3k test, over 80 words.

Train	Test
$p_1 \equiv p_2$	$p_2 \wedge p_7$
$p_1 \sqsupset p_5$	$p_2 \sqsupset p_5$
$p_4 \sqsupset p_8$	~~$p_5 = p_6$~~
$p_5 \mid p_7$	~~$p_7 \sqsubset p_4$~~
$p_7 \wedge p_1$	$p_8 \sqsubset p_4$

	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)**

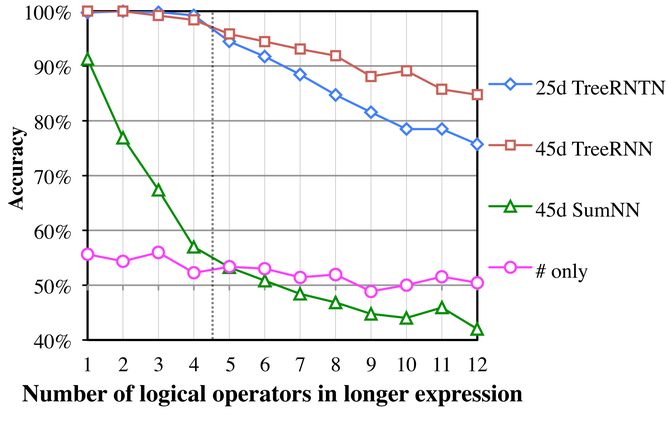
Figures are reported as % accuracy and (macro-averaged F1)

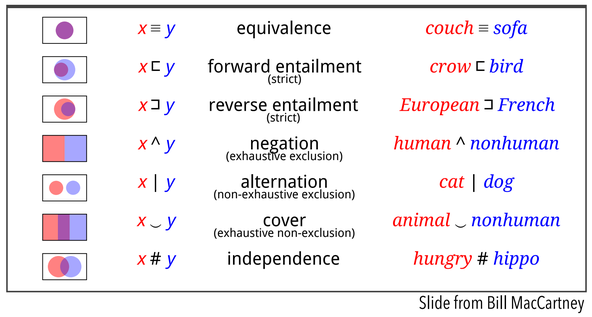
Learning recursive, functional definitions

- Phrase and sentence meanings are built compositionally out of shorter phrases and sentences following a recursive structure.
- Testing ability to learn to handle recursive structure: we train a model on short sentences and test it on longer ones.
- Data: Statements of propositional logic, 60k short training examples, 12k training examples of up to triple the length.
- NB: Model must compare statements with unvalued variables, more in common with 3-SAT than plain Boolean evaluation.

not p_3	^	p_3
not not p_6	≡	p_6
p_3	⊏	(p_3 *or* p_2)
(p_1 *or* (p_2 *or* p_4))	⊐	(p_2 *and not* p_4)
not (*not* p_1 *and not* p_2)	≡	(p_1 *or* p_2)

Short examples. All logical symbols (proposition variables, *and*, *or*, and *not*) are treated as words and have learned embedding vectors.





Slide from Bill MacCartney

Monotonicity reasoning and quantifiers

- Monotonicity + quantification are a classic case study for formal semantics: If *all dogs bark*, do *all animals make sounds*?
- Artificial data with a 20 word vocabulary:

Train:	*(most turtle) swim*	|	*(no turtle) move*
	(all lizard) reptile	⊏	*(some lizard) animal*
Test:	*(most turtle) reptile*	|	*(all turtle) (not animal)*

	Train	Test
# only	35.4 (7.5)	35.4 (7.5)
25d SumNN	96.9 (97.7)	93.9 (95.0)
25d TreeRNN	99.6 (99.6)	99.2 (99.3)
25d TreeRNTN	**100.0 (100.0)**	**99.7 (99.5)**

Figures are reported as % accuracy and (macro-averaged F1)

Can NNs learn to do inference over real English?

- Train/test on SICK entailments (4.5k training examples).
- Best purely-learned system to date, but even with words from GloVe, noisy extra data from DenotationGraph, and significant preprocessing, accuracy still below the SotA (77% vs. 85%).
- 4.5k examples is not enough to learn English compositional semantics. Need more data and better unsupervised methods.

Compositional neural network models for natural language meaning already do well on phenomena like semantic similarity and sentiment that engage the strengths of these models' continuous vector representations. Our artificial data experiments find no fundamental obstacles to also being able to learn representations capable of modeling formal semantic notions of meaning composition from scratch, given enough data.