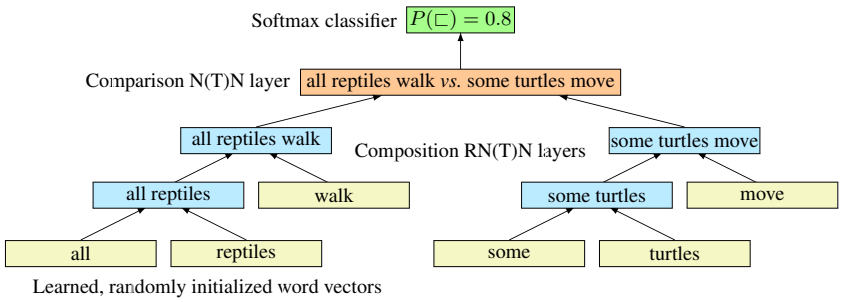
Recursive Neural Networks for Learning Logical Semantics

Samuel R. Bowman[*†] sbowman@stanford.edu
Christopher Potts[*] cgpotts@stanford.edu
Christopher D. Manning[*†‡] manning@stanford.edu

[*]Stanford Linguistics
[†]Stanford NLP Group
[‡]Stanford Computer Science

Supervised recursive neural network models (RNNs) for sentence meaning have been successful in an array of sophisticated language tasks, but it remains an open question whether they can learn compositional semantic grammars that support logical deduction. We address this question directly by for the first time evaluating whether each of two classes of neural model — plain RNNs and recursive neural tensor networks (RNTNs) — can correctly learn relationships such as entailment and contradiction between pairs of sentences, where we have generated controlled data sets of sentences from a logical grammar. Our first experiment evaluates whether these models can learn the basic algebra of logical relations involved. Our second and third experiments extend this evaluation to complex recursive structures and sentences involving quantification. We find that the plain RNN achieves only mixed results on all three experiments, whereas the stronger RNTN model generalizes well in every setting and appears capable of learning suitable representations for natural language logical inference.

Recursive neural network models



In our experiments, we train pairs of recursive (tree structured) neural network models [1] which are joined together with a shared top layer that generates features for a classifier. The classifier predicts the logical relation that holds between the sentences represented by the two trees (entailment in the above; the table below reviews the full inventory of relations we predict). For an activation function, we use either a plain NN layer or a tensor combination layer.

Inference and the semantic relations

Our aim is to evaluate the ability of learned recusive models to represent semantic structure. We pursue this through an inference task, where the models must learn to choose the logical relation that applies between a pair of statements. The possible relations are the seven below from [2]. This is distinct from the limited prior work on learning neural network models for formal semantics [3, 4], which use an interpretation task in which the model evaluates sentences as *true* or *false* with respect to some representation of the world. Our approach allows us to sidestep serious open problems involved in the representation of a complete set of knowledge about the world.

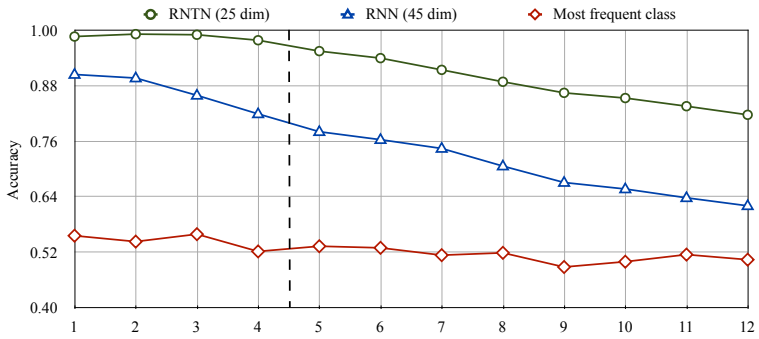
Name	Symbol	Set-theoretic definition	Example
entailment	$x \sqsubset y$	$x \subset y$	*turtle, reptile*
reverse entailment	$x \sqsupset y$	$x \supset y$	*reptile, turtle*
equivalence	$x \equiv y$	$x = y$	*couch, sofa*
alternation	$x \mid y$	$x \cap y = \emptyset \wedge x \cup y \neq \mathcal{D}$	*turtle, warthog*
negation	$x \,^\wedge\, y$	$x \cap y = \emptyset \wedge x \cup y = \mathcal{D}$	*able, unable*
cover	$x \smile y$	$x \cap y \neq \emptyset \wedge x \cup y = \mathcal{D}$	*animal, non-turtle*
independence	$x \# y$	(else)	*turtle, pet*

Reasoning with atomic symbols

If any model is to learn the behavior of a relational logic like the one presented here from a finite amount of data, it must learn to deduce new relations from already seen relations. Our first experiment evaluates the ability of our models to do this over pairs of atomic symbols. The model is trained on a randomly generated set of consistent statements like $\{a \sqsubset b, b \sqsubset c, c \,^\wedge\, d\}$, and tested on novel examples that follow from the statements seen in training, like $\{a \sqsubset c\}$. A tuned RNTN reached greater than 99% test accuracy, but no plain RNN surpassed 89%.

Recursive structure in propositional logic

Our second experiment introduces compositional structure to our examples, replacing the atomic statements above with pairs of short statements of propositional logic, like *not* $a \mid ((a \ (\textit{and}\ b)) \ (\textit{and}\ c))$. We train our models on only pairs of statements with up to four symbols (corresponding to test accuracy figures to the left of the dashed line below), but observe that the RNTN performs reasonably both on pairs of that length and on much longer test pairs.



Reasoning with natural language quantifiers and negation

For our third experiment, we generate pairs of sentences in which each sentence contains one quantifier, and any of a small set of common nouns, as in the example *(no warthogs) move* $\sqsubset$ *(no (not reptiles)) swim*. The parentheses indicate the tree structure for each sentence as it will be used by the model. We defined several different types of train–test split for this experiment. RNTNs performed either well ($> 85\%$ accuracy) or perfectly on all of them, while plain RNNs did not break 80% in any setting. These experiments differentiate the increased power of RNTNs better than previous work and provide the most convincing demonstration to date of the ability of neural networks to model semantic inferences in complex natural language sentences.

References

[1] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, 2013.

[2] W. MacCartney and C. D. Manning. An extended model of natural logic. In *Proceedings of IWCS*, 2009.

[3] E. Grefenstette. Towards a formal distributional semantics: Simulating logical calculi with tensors. *arXiv preprint 1304.5823*, 2013.

[4] T. Rocktäschel, M. Bosnjak, S. Singh, and S. Riedel. Low-dimensional embeddings of logic. *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, 2014.