Towards Compositional Learning for Natural Language Understanding

Master of Technology Thesis

Arjun Verma
IMT2017008

Prof. Gopalakrishnan Srinivasaraghavan
Tim Klinger

10th June, 2022

Outline

- Motivation
  - Fundamental Problems with Deep Learning
  - Compositionality
  - Compositionality in Natural Language
  - Compositional Skills of Deep Learning
- Proposed Methodology
  - Dataset Preparation
  - Architectural Setup
- Experimental Results
  - Metric and Configuration
  - Main Results
- Future Work

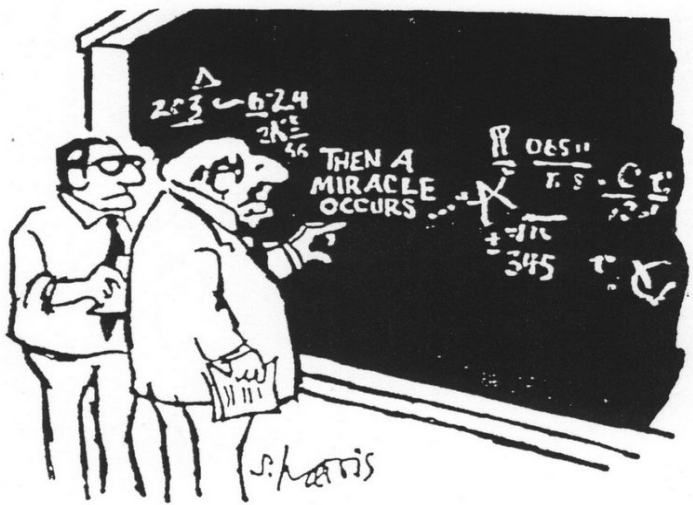
Deep Learning

What it's not : A voodoo black-boxed end-all-be-all tool that it is often notoriously proclaimed to be.

What it is : A powerful statistical technique that is capable of recognizing patterns based on adequate sample data.

Breakthrough Performances

- Vision ⟷ YOLO, DINO
- Language ⟷ BERT
- Speech ⟷ Conformer
- RL ⟷ AlphaZero



I think you should be a little more specific, here in Step 2

Image taken from
https://analyticsindiamag.com/deep-learning-is-not-so-much-a-black-box-anymore-and-thats-a-great-development/

Fundamental Problems with Deep Learning

Extremely Data Hungry

Cognitive Perspective

- Humans possess the ability to learn from a single to few examples, DL requires thousand of examples to attain similar efficiency
- Even 7 month old infants have the capability to pick up abstract language-like rules from very few unlabelled examples [1]

Applicability Perspective

- Massive resources for critical domains with low margin of errors
- Incoherent Replicability

[1] G.F. Marcus, S Vijayan, Shoba Bandi-Rao, and Peter Vishton. Rule learning by seven-month-old infants. Science (New York, N.Y.), 283:77–80, 02 1999.

Fundamental Problems with Deep Learning

Lack of an Innate Hierarchical Bias

- Linguists have concretely established that natural language is hierarchical and yet most language models treat sentences as a mere sequence of words.
- DL learns from a “flat” set of features. In a simple, non-hierarchical list, every element intrinsically holds equal weightage.
- Attempts at capturing hierarchies have mostly been superficial such as appending sequential positionings [2]

Lack of an hierarchical bias is a key factor in DL’s lack of compositional skills

[2] Measuring compositional generalization: A comprehensive method on realistic data, 2019.

Compositionality

Complex structures in nature are often recursively constructed out of smaller primitive structures

It further argues that structures are composed hierarchically from elementary substructures utilizing a set of production rules.

What does this imply for DL?

- Compositionality encourages the fact that substructures and production rules may be learnt from a finite amount of data.
- The learnt substructures and rules could then be used to generalize across different combinatorial scenarios.

Compositionality in Natural Language

Natural language is highly productive. It is considered to be an “infinite employment of finite means”

Linguists have widely attributed this productivity to the principle of semantic compositionality.

> “The meaning of a (syntactically complex) whole is a function only of the meanings of its (syntactic) parts together with the manner in which these parts were combined.” [4]

This could also be mathematically formulated as [3],

$$m(f(e_1, ..., e_k)) = g(m(e_1), ..., m(e_k))$$

, where f represents a syntactic operation
g represents a semantic operation
m represents a meaning function

[4] Francis Jeffry Pelletier. The principle of semantic compositionality. Topoi, 13(1):11–24, Mar 1994.

[3] Chenyao Liu et al. Learning algebraic recombination for compositional generalization, 2021

Compositional Skills of Deep Learning

Do They Possess It?

- For natural language, the authors of [5] have shown that while DL architectures generalize well when the training and test set distributions are similar, they fail spectacularly when these distributions are intentionally diverged.

Train set	Test set
Who directed Inception?	Did Greta Gerwig direct Goldfinger?
Did Greta Gerwig produce Goldfinger?	Who produced Inception?
...	...

- Even in Reinforcement Learning, very rarely do algorithms generalize their abstract plans in unseen scenarios. Transfer learning in across Atari games is still considered to be difficult [6]

[5] Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. 2017

[6] Girish Joshi and Girish Chowdhary. Cross-domain transfer in reinforcement learning using target apprentice, 2018.

Compositional Skills of Deep Learning

Are They Trying to Possess It? (Considering only natural language)

- Extensive studies [7],[8] demonstrate that recurrent networks are not simply relying on surface heuristics but are actually building pseudo-syntactic processing mechanisms

1. Colorless green ideas sleep furiously
2. Colorless sleep ideas green furiously

- They show promise in differentiating between grammatical and ungrammatical nonsensical sentences which indicates an attempt at productivity through compositionality.
- Recent work by [9] has also hinted at transformer-based models' ability to achieve compositionality ….

112 million data points and 49 million parameters later

[7] Marco Baroni. Linguistic generalization and compositionality in modern artificial neural networks. Philosophical Transactions of the Royal Society B: Biological Sciences, 2020.

[8] Tal Linzen and Marco Baroni. Syntactic structure from deep learning. Annual Review of Linguistics, 2021.

[9] Jacob Russin, et al. Compositional processing emerges in neural networks solving math problems, 2021.

Key Takeaway

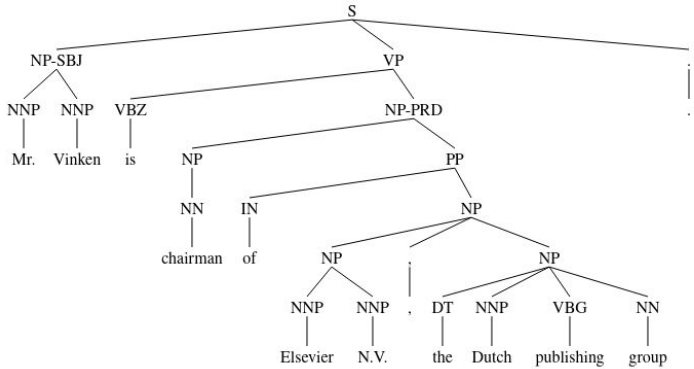
The argument thus goes that achieving compositionality is not the end goal. Rather, it is to come up with useful inductive biases that would help DL models attain compositional skills in a data efficient manner.

Proposed Methodology

Dataset

The Penn Treebank (PTB)

- We will be using the principle of compositionality to combine syntactic and semantic elements of natural language together.
- For our syntactic needs, we utilize the Penn Treebank. It is one of the largest human annotated dataset comprising of parse trees for 49,208 English sentences.
- As the dataset is behind a paywall, we use NLTK to get 5% of the PTB. These total to 3794 sentences.



Dataset Modification

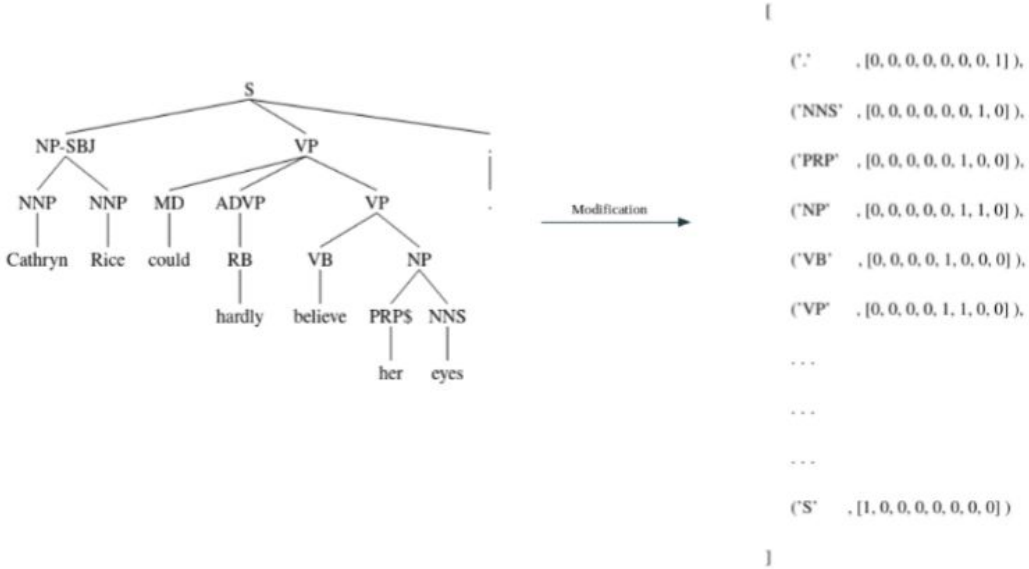


Figure FC3.3: Given the sample input : "Cathryn Rice could hardly believe her eyes.", we demonstrate the first six steps of the transformation list obtained after our modification procedure. As mentioned earlier, the parse tree on the left has been obtained from the PTB. Each tuple in the list represents a transformation step. The first element of a tuple is the abstraction category while the second element denotes the span of words being abstracted. Please note, for an ease of comprehension, we have left the abstraction category as is. In the actual dataset, they are represented as one-hot encodings. This transformation list serves as the target in our (input, target) pairs.

Model : Recurrent Independent Mechanisms (RIMs)[10]

Inspiration

- The world is governed by physical laws and has underlying structure.
- Isolated rules or independent mechanisms are often sufficient to explain variation in a subsystem
- Most mechanisms do not interact highly with each other.
- Only a small subset of all mechanisms are relevant for any particular problem of interest.

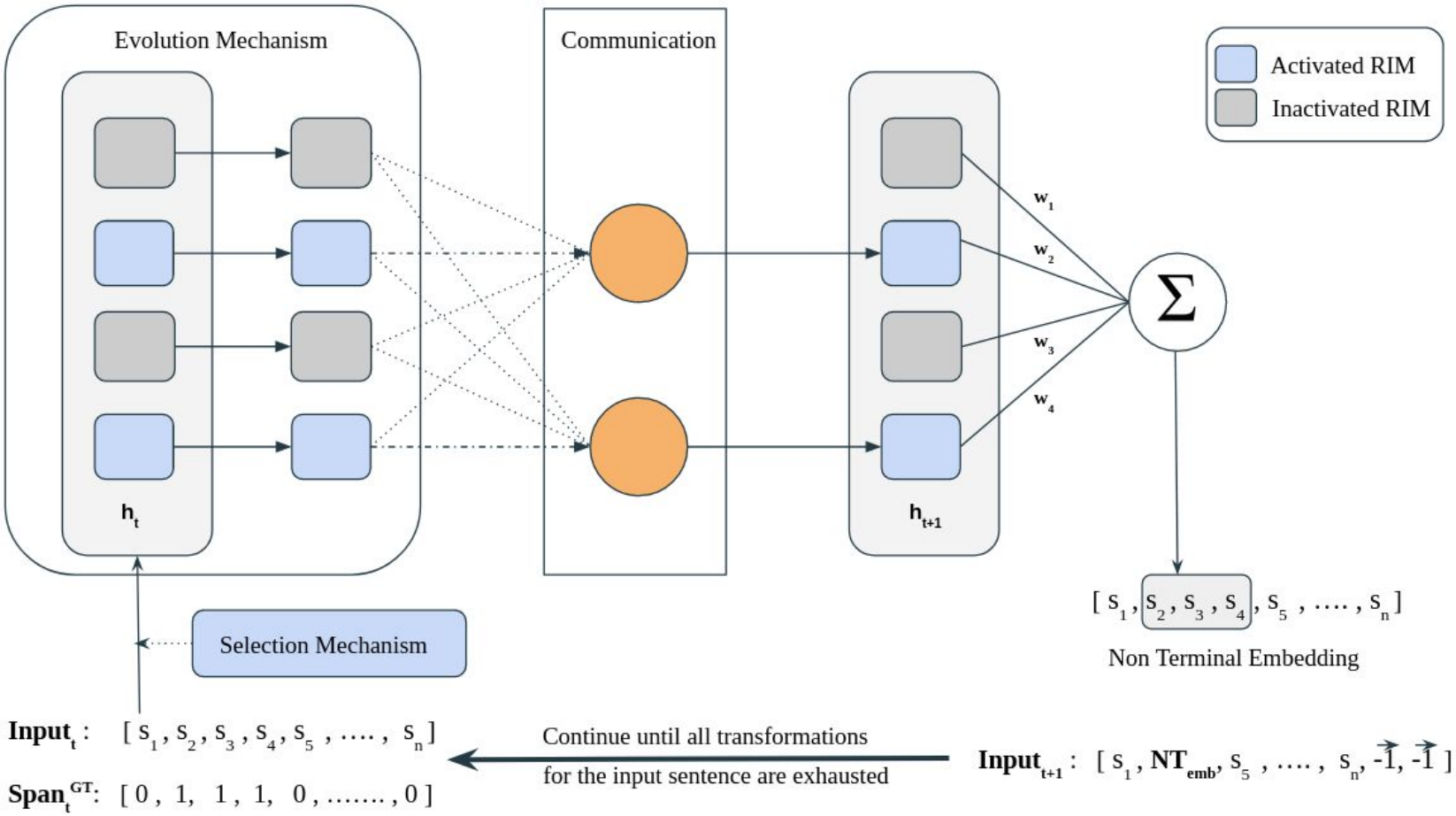
[10] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Scholkopf. Recurrent independent mechanisms, 2019.

Model : Recurrent Independent Mechanisms (RIMs)

Broad Idea

- Design an overall system which consists of independent subsystems evolving over time.
- Not all systems evolve at all times, only subsystems displaying a significant overlap in their interactions are considered.
- Such an architecture favoring modularity and dynamic recombination could be quite beneficial over their monolithic counterparts, when it comes to grasping compositional structures.

Architectural Setup



Model Step

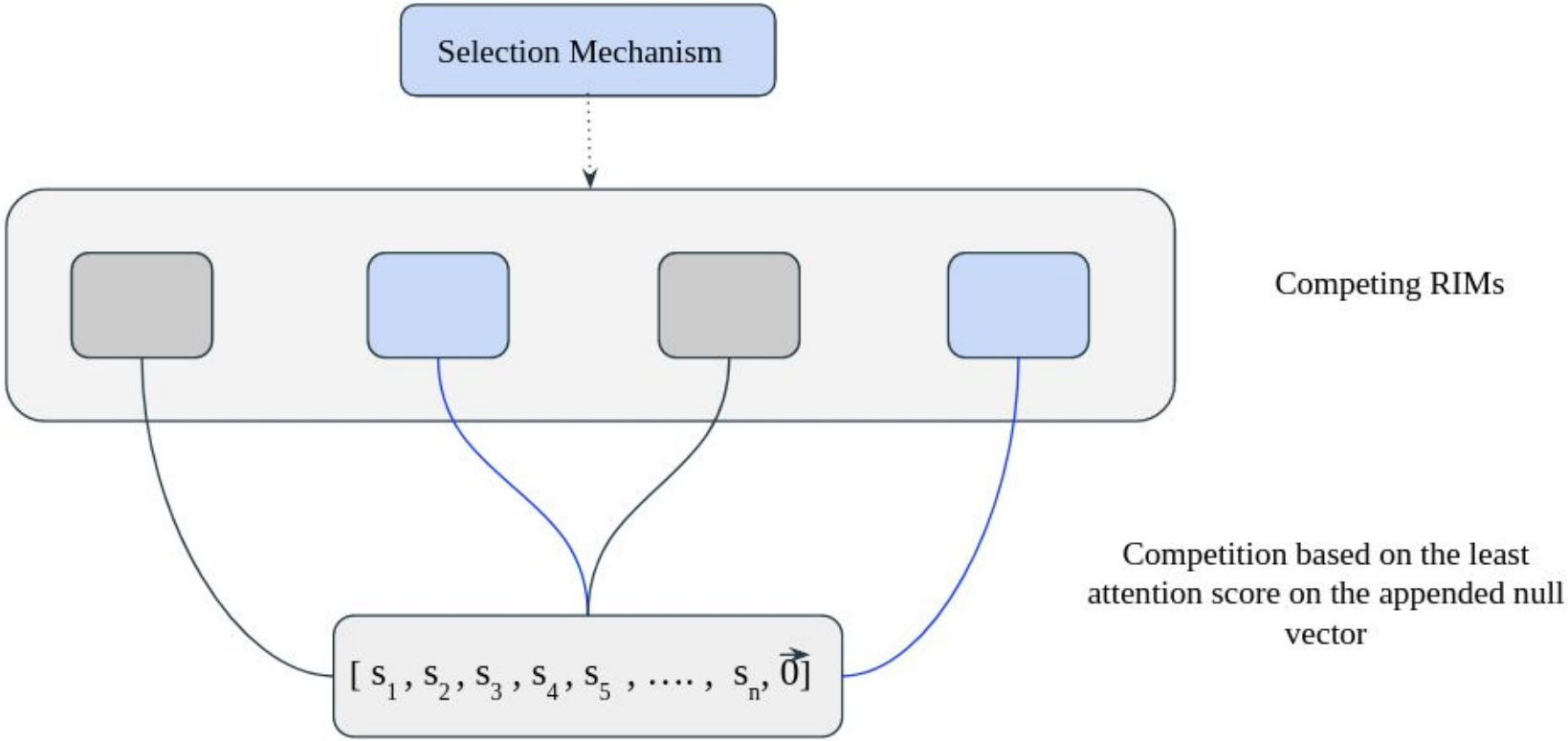
1. Selection Mechanism

- Append a null vector to the incoming input sequence
- Produce a linear projection from each RIM unit's hidden state h_t to act as a query
- Project the set of inputs at that time step to produce a matrix of keys and values
- Compute attention for each RIM unit,

$$A_k^{(in)} = sigmoid\left(\frac{h_t W_k^q (XW^e)^T}{\sqrt{d_e}}\right) XW^v$$

where $\theta_k^{(in)} = (W_k^q, W^e, W^v)$ represents the parameters for k^{th} RIM unit and A^k are the attention scores computed over all the elements in X for the k^{th} RIM unit.

- The k_A RIM units with the smallest attention weights on null vector are considered active at that time step


Selection Mechanism
Competing RIMs
Competition based on the least attention score on the appended null vector
[s_1 , s_2 , s_3 , s_4 , s_5 , …. , s_n, $\vec{0}$]

Model Step

2. Evolution Mechanism

Each RIM unit is essentially a LSTM. To update the dynamics after a step, we need to compute new hidden states for each of the LSTM. Let S_t to be the set of activated units,

For inactive RIM units,

$$h_{t+1,k} = h_{t,k}, \quad \forall k \notin S_t$$

For active RIM units,

$$h_{t+1,k} = LSTM(A_k^{(in)}, h_{t,k}; \quad \theta_k), \quad \forall k \in S_t$$

where $A_k^{(in)}$ represents the attended inputs for the k^{th} RIM and θ_k represents the parameters of the k^{th} RIM.

Model Step

3. Communication Mechanism

- For each active RIM unit, project the hidden state to produce a query
- Project the hidden states of all other units, to produce keys and values
- Now, compute the attention scores and add it to the residual,

$$K_{t,k} = \tilde{W}_k^e h_{t+1,k} \quad , \quad \forall k$$

$$V_{t,k} = \tilde{W}_k^v h_{t+1,k} \quad , \quad \forall k$$

$$Q_{t,k} = \tilde{W}_k^q h_{t+1,k} \quad , \quad \forall k \in S_t$$

$$h_{t+1,k} = softmax(\frac{Q_{t,k}(K_{t,:})^T}{\sqrt{d_e}})V_{t,:} \quad + \quad h_{t+1,k} \quad , \quad \forall k \in S_t$$

- Disallow backpropagation through inactive RIM units

Model Training

1. Given an (input, target) we begin by feeding in the input sequence at t = 0. The input sequence is a vector of word embeddings. The word embeddings are 100-d GloVe embeddings.

2. Generate a weighted attention vector from the model indicating the span it is currently focusing on,

$$attn_{combined} = \Sigma_k^R w_k \cdot attn_{1:n}^k$$

where, R indicates the total number of RIM units and $attn_{1:n}^k$ represents the attention scores for the k^{th} RIM unit over the n words in the sentence.

3. The weights are generated as,

$$w_k = 0 \quad , \quad \forall k \notin S_t$$

$$w_k = softmax(1 - attn_{null\_vec}^k) \quad , \quad \forall k \in S_t$$

where, $attn_{null\_vec}^k$ indicates the attention score over the appended null vector

Model Training

4. Compute loss using the ground truth span and the weighted attention vector,

$$Loss^t = BinaryCrossEntropy(attn^t_{combined}, span^t_{transform})$$

where, $span^t_{transform}$ is the ground truth span obtained from the step at time t

5. Generate a non-terminal embedding for the span using the same weights as before,

$$N.T_{embed} = \Sigma^R_k w_k \cdot h^k_{t+1}$$

where, R indicates the total number of RIM units and h^k_{t+1} represents the newly generated hidden state for k^{th} RIM unit.

6. Insert the generated non-terminal embedding in place of the span and appropriately pad the new sequence with vector of -1's to keep the dimensionality consistent.

Expected Outcomes

1. Given our learning setup, we can test whether the RIM model is able to reasonably generalize on the task of grammar induction.

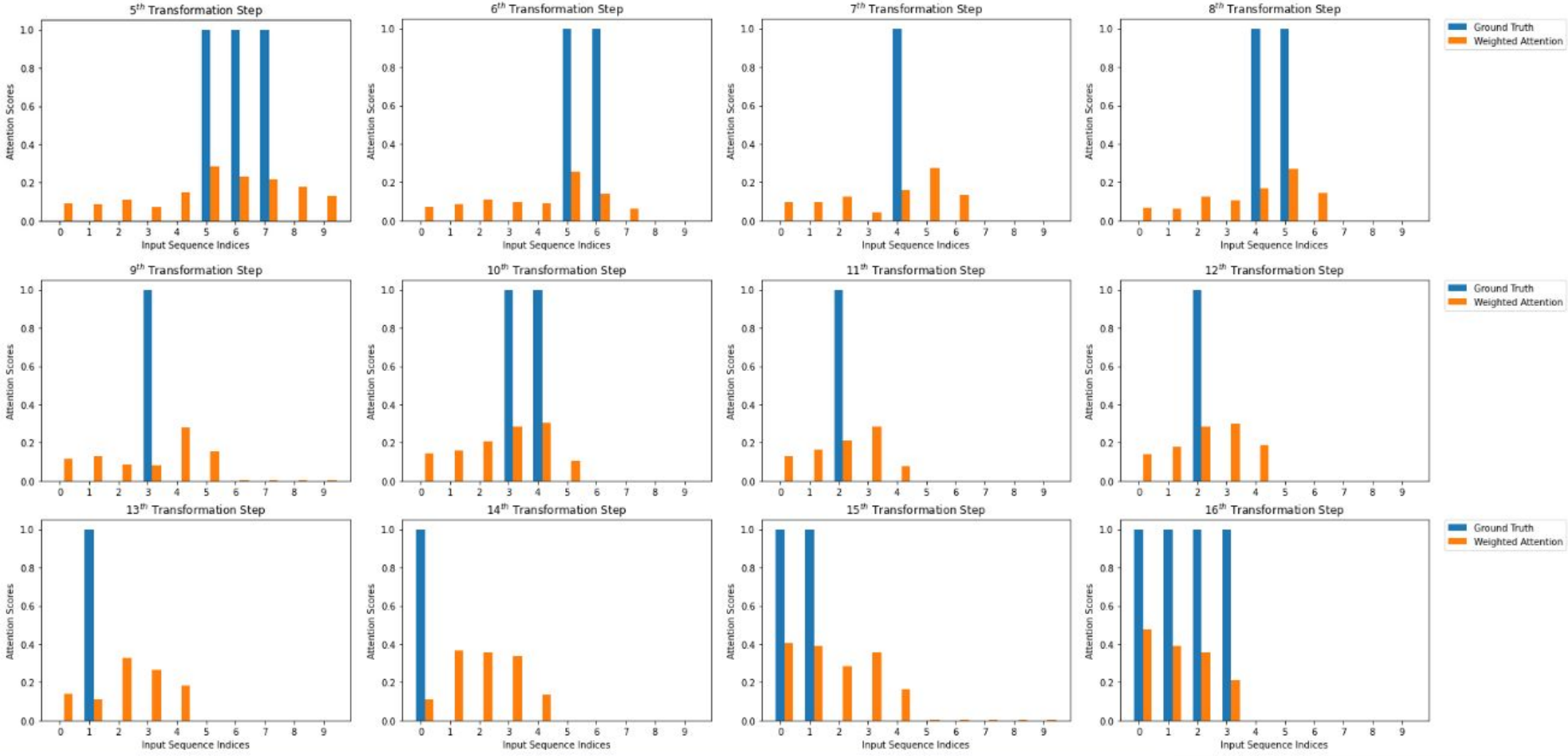
2. If the above hypothesis holds, we potentially have the ability to make compositional embeddings for sentences. We can test the effectiveness of these embeddings on classification tasks such as sentiment analysis.

3. Again, if the first hypothesis holds, we can now abstract out sentences with each transformational step and can make compositional predictions on downstream tasks.

Within the scope of this thesis, we shall only be focusing on the grammar induction capabilities of the model.

Experimental Results

Visualizing Outputs



Defining Metrics

Standard Metrics

Standard metric for evaluating a phrase structure parser is *bracket score*.

- Precision: Out of all brackets that the parser detects, how many are also present in the gold standard?
- Recall: Out of all brackets in the gold standard, how many does the parser also detect?
- F1-Score: It is the harmonic mean between precision and recall,

$$\text{F1-score} = (2 * precision * recall) / (precision + recall)$$

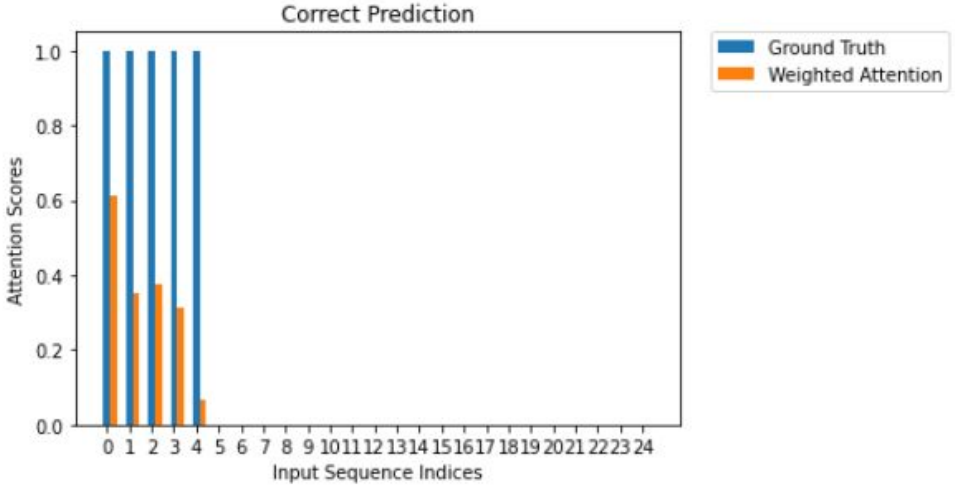
Our Metrics

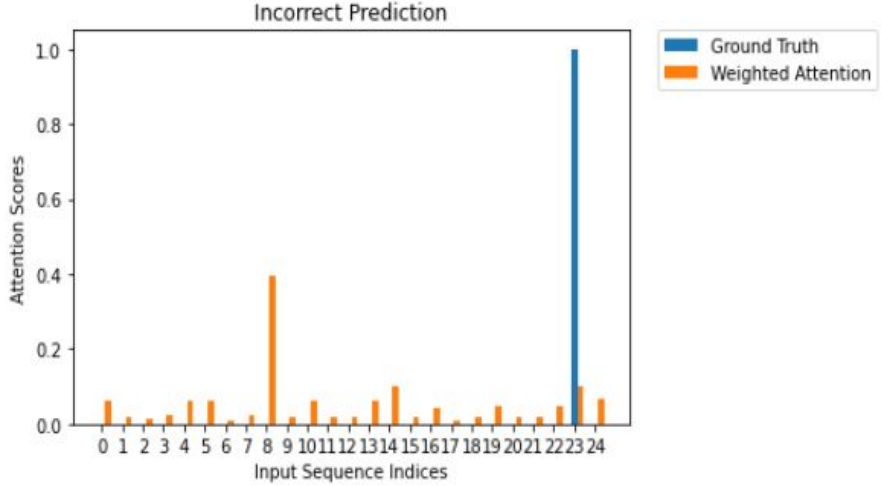
As we are working with only 5% of the PTB, our model is barely capable of being tested on such stringent metrics.

To check whether our model is at least proceeding in the correct direction, we come up with a more lenient metric to evaluate its performance,

- If our model has assigned the unique highest attention score within the indices representing the ground truth span, we take the prediction to be correct.
- We also allow a relaxation window of size 2 on either sides of the span.

Example : If at a particular step, words at indices [4,5] were to be combined, our model would be adjudged accurate if it gives the unique highest attention score between [2,7]


Correct Prediction
Ground Truth
Weighted Attention
Attention Scores
0.0
0.2
0.4
0.6
0.8
1.0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Input Sequence Indices

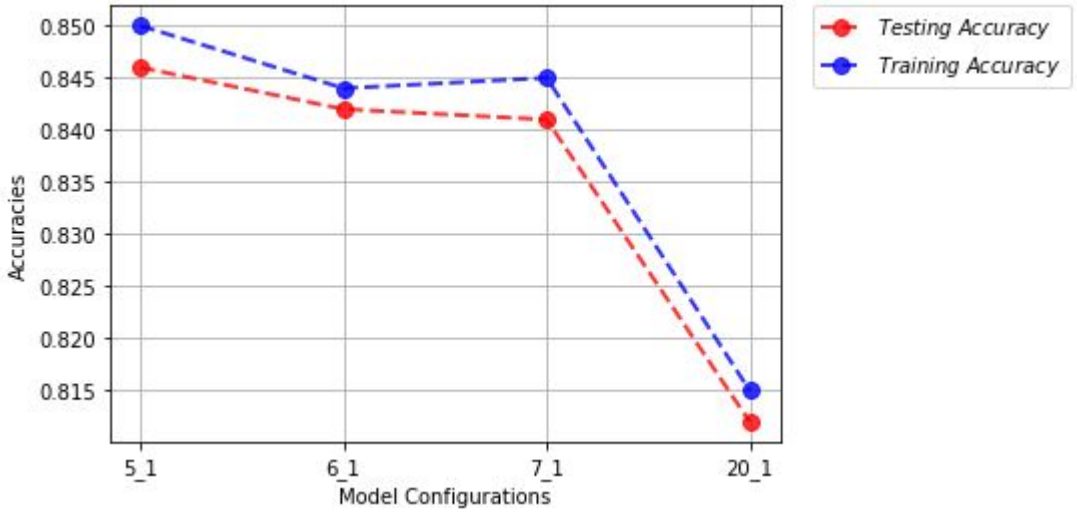

Incorrect Prediction
Ground Truth
Weighted Attention
Attention Scores
0.0
0.2
0.4
0.6
0.8
1.0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Input Sequence Indices

Model Configuration

- Train - Test split of 75% - 25%
- 5 RIM Units and Top-3 Selection Mechanism
- Trained for 11 epochs
- Reduce LR on Plateau Scheduler
  - Initial rate : 2e-5
  - Patience : 3
  - Threshold : 1e-3
  - Factor : 0.5

Configurational Experimentation

- No thumb-of-rule patterns to help us with our choices
- One intuitive pattern that could certainly be expected is of Top-1 selection mechanism. Allowing only one heuristic to operate should almost always be inefficient.



Main Results

Table TC4.1: Accuracies on individual syntactic tags

Model	NP	VP	PP	SBAR	ADJP	ADVP	Sentence
1-Layered (5, 3)	93.28	96.43	95.89	98.33	93.45	87.08	86.54

- The scores have been computed as, (correct / total_occurences)
- They have been averaged over 5 runs of the model.
- Traditionally, parser metrics are computed on 6 major syntactic tags :
  NP, VP, PP, SBAR, ADJP and ADVP
- The sentence level metric takes into account all possible non-terminals (53 in number)

Case for Allowed Window

Table TC4.2: Comparing single and multiple word abstraction accuracies by varying window sizes

Window Size	Single-Word Spans	Multiple-Word Spans
0	6	85
1	61	93
2	79	96

Further analysis shows that the model performs well when multiple words are to be combined. Converging on single word spans is a difficult task for the model at the moment.

Case for Deeper Layers and More Data

Table TC4.3: Comparing a deeper configuration

Model	NP	VP	PP	SBAR	ADJP	ADVP	Sentence
1-Layered (5, 3)	93.28	96.43	95.89	98.33	93.45	87.08	86.54
2-Layered (5, 3)	96.51	98.60	97.8	100.0	96.1	93.44	87.23

Table TC4.4: Comparing span scores (window size 0)

Category	1-Layered (11 epochs)	2-Layered (11 epochs)	2-Layered (20 epochs)
Single-Word	~ 6	~ 20	~ 15
Multi-Word	~ 89	~ 89	~ 91

- Initial results hint that more data might improve the model's performance.
- Additional strategies might be needed.

Future Work

- Improve performance on the grammar induction task till span selection overlaps better with ground truth spans.

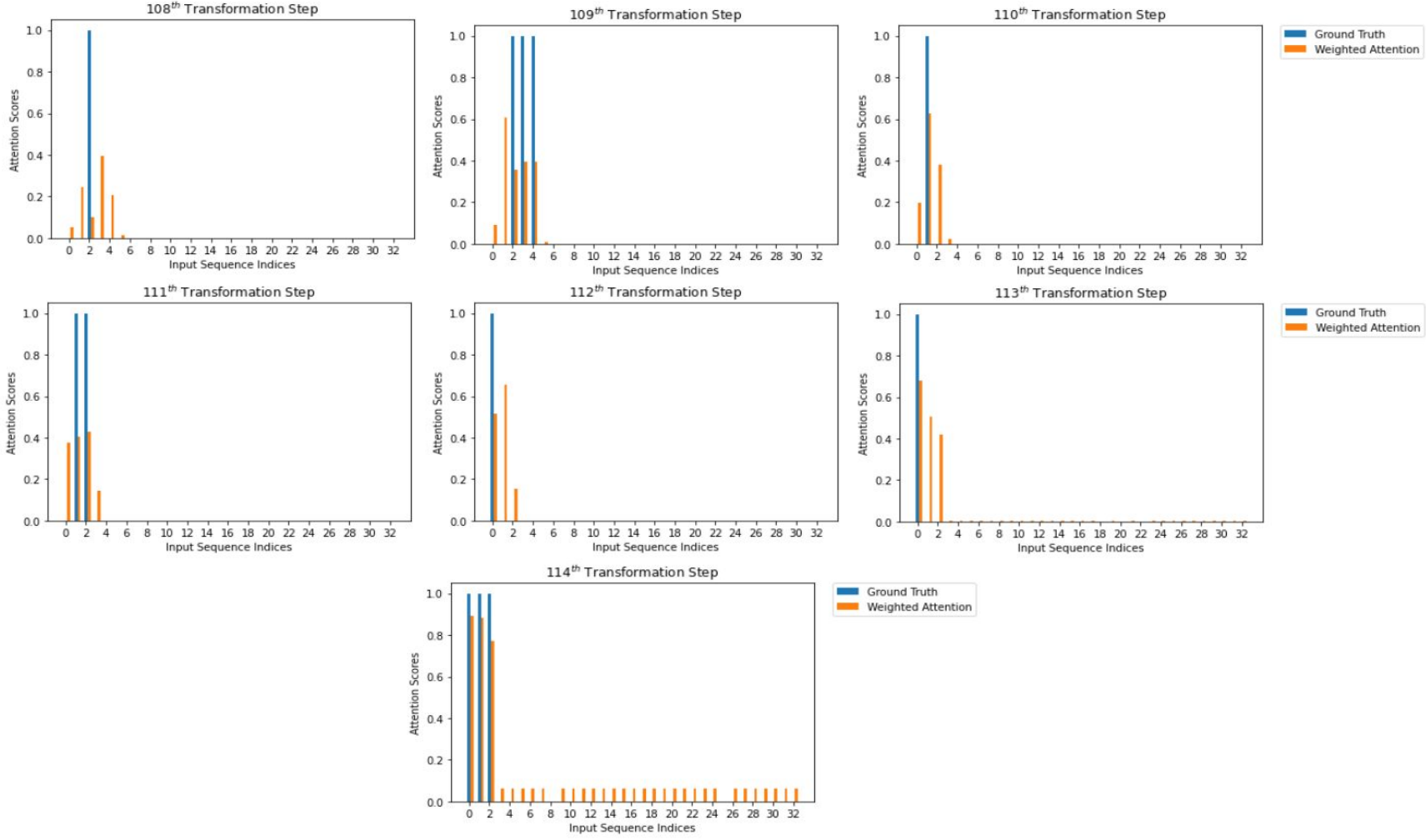
- Test the efficiency of compositional embeddings being generated as part of the induction process by tuning the model to downstream tasks such as sentiment analysis.

Backup

1. CC	Coordinating conjunction	25. TO	*to*
2. CD	Cardinal number	26. UH	Interjection
3. DT	Determiner	27. VB	Verb, base form
4. EX	Existential *there*	28. VBD	Verb, past tense
5. FW	Foreign word	29. VBG	Verb, gerund/present participle
6. IN	Preposition/subordinating conjunction	30. VBN	Verb, past participle
7. JJ	Adjective	31. VBP	Verb, non-3rd ps. sing. present
8. JJR	Adjective, comparative	32. VBZ	Verb, 3rd ps. sing. present
9. JJS	Adjective, superlative	33. WDT	*wh*-determiner
10. LS	List item marker	34. WP	*wh*-pronoun
11. MD	Modal	35. WP$	Possessive *wh*-pronoun
12. NN	Noun, singular or mass	36. WRB	*wh*-adverb
13. NNS	Noun, plural	37. #	Pound sign
14. NNP	Proper noun, singular	38. $	Dollar sign
15. NNPS	Proper noun, plural	39. .	Sentence-final punctuation
16. PDT	Predeterminer	40. ,	Comma
17. POS	Possessive ending	41. :	Colon, semi-colon
18. PRP	Personal pronoun	42. (	Left bracket character
19. PP$	Possessive pronoun	43.)	Right bracket character
20. RB	Adverb	44. "	Straight double quote
21. RBR	Adverb, comparative	45. ‘	Left open single quote
22. RBS	Adverb, superlative	46. “	Left open double quote
23. RP	Particle	47. ’	Right close single quote
24. SYM	Symbol (mathematical or scientific)	48. ”	Right close double quote

1.	ADJP	Adjective phrase
2.	ADVP	Adverb phrase
3.	NP	Noun phrase
4.	PP	Prepositional phrase
5.	S	Simple declarative clause
6.	SBAR	Clause introduced by subordinating conjunction or *0* (see below)
7.	SBARQ	Direct question introduced by *wh*-word or *wh*-phrase
8.	SINV	Declarative sentence with subject-aux inversion
9.	SQ	Subconstituent of SBARQ excluding *wh*-word or *wh*-phrase
10.	VP	Verb phrase
11.	WHADVP	*wh*-adverb phrase
12.	WHNP	*wh*-noun phrase
13.	WHPP	*wh*-prepositional phrase
14.	X	Constituent of unknown or uncertain category
	Null elements	
1.	*	“Understood” subject of infinitive or imperative
2.	0	Zero variant of *that* in subordinate clauses
3.	T	Trace—marks position where moved *wh*-constituent is interpreted
4.	NIL	Marks position where preposition is interpreted in pied-piping contexts

	PRPN	ON	PCFG	Comp. PCFG
Gold	47.3	48.1	50.8	55.2
Left	1.5	14.1	11.8	13.0
Right	39.9	31.0	27.7	28.4
Self	82.3	71.3	65.2	66.8
SBAR	50.0%	51.2%	52.5%	56.1%
NP	59.2%	64.5%	71.2%	74.7%
VP	46.7%	41.0%	33.8%	41.7%
PP	57.2%	54.4%	58.8%	68.8%
ADJP	44.3%	38.1%	32.5%	40.4%
ADVP	32.8%	31.6%	45.5%	52.5%


108th Transformation Step
109th Transformation Step
110th Transformation Step
111th Transformation Step
112th Transformation Step
113th Transformation Step
114th Transformation Step
Attention Scores
Input Sequence Indices
Ground Truth
Weighted Attention

RIMs for Language

1. RIM units as language acquisition heuristics.

2. Sigmoidal Attention due to the conditional independence property of Factored-MDPs

3. Losses on attention vector as an MoE solution.