On Learning the Past Tenses of Verbs

Rumelhart, McClelland

1985

Big Picture

How do we (humans) use and acquire knowledge of language?

Two competing ideas:

1. *Explicit, inaccessible rule view*: rules of language are stored in explicit form
2. *Connectionist models:* capture “rule-like” behavior with no explicit form of rules

History

- First connectionist implementation by Rumelhart & McClelland in 1986
  - Number of criticisms:
    - Error rate on “unseen” verbs is high -> Do these models reach adult competence?
    - Pinker and Prince (1988) and Lachter and Bever (1988): Extremely poor empirical performance
- Improved results by MacWhinney & Leinbach in 1991, replaced Wickelfeature representation with UNIBET
- Resurgence of neural networks today
  - Kirov and Cotterell (2018) show that the Encoder-Decoder network architectures preclude many of P&P’s arguments

Three claims from R&M connectionist model

1. The model captures the U-learning three-stage pattern of acquisition.
2. The model captures most aspects of differences in performance on different types of regular and irregular verbs.
3. The model is capable of responding to regular and irregular verbs seen in training *and* low frequency “unseen” verbs.

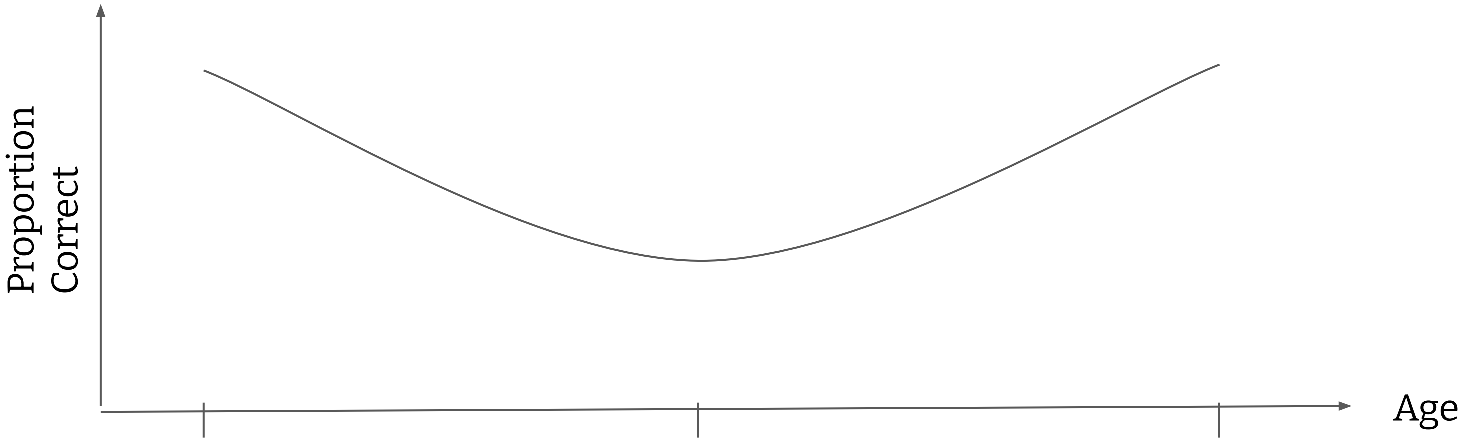
R&M argument

1. The model demonstrates that it can acquire past tense without rules. So, "[t]he child need not figure out what the rules are, nor even that there are rules. The child need not decide whether a verb is regular or irregular."
2. If no explicit rules, why should children generate forms that they have never heard of?

   "They do so because the past tenses of similar verbs they are learning show such a consistent pattern that the generalization from these similar verbs outweighs the relatively small amount of learning that has occured on the irregular verb question."

Discussion

U-learning three-stage pattern of past tense acquisition



Stage 1
Rote learning of regular verbs

Initial error free performance

Stage 2
Rule extraction, few new irregular verbs

Over-regularization errors

Stage 3
Regular and irregular verbs

Recovery from errors

Train:
10 trials, 10 high-frequency verbs

190 more trials, 410 medium-frequency verbs

Test:
86 low-frequency verbs

Connectionist model

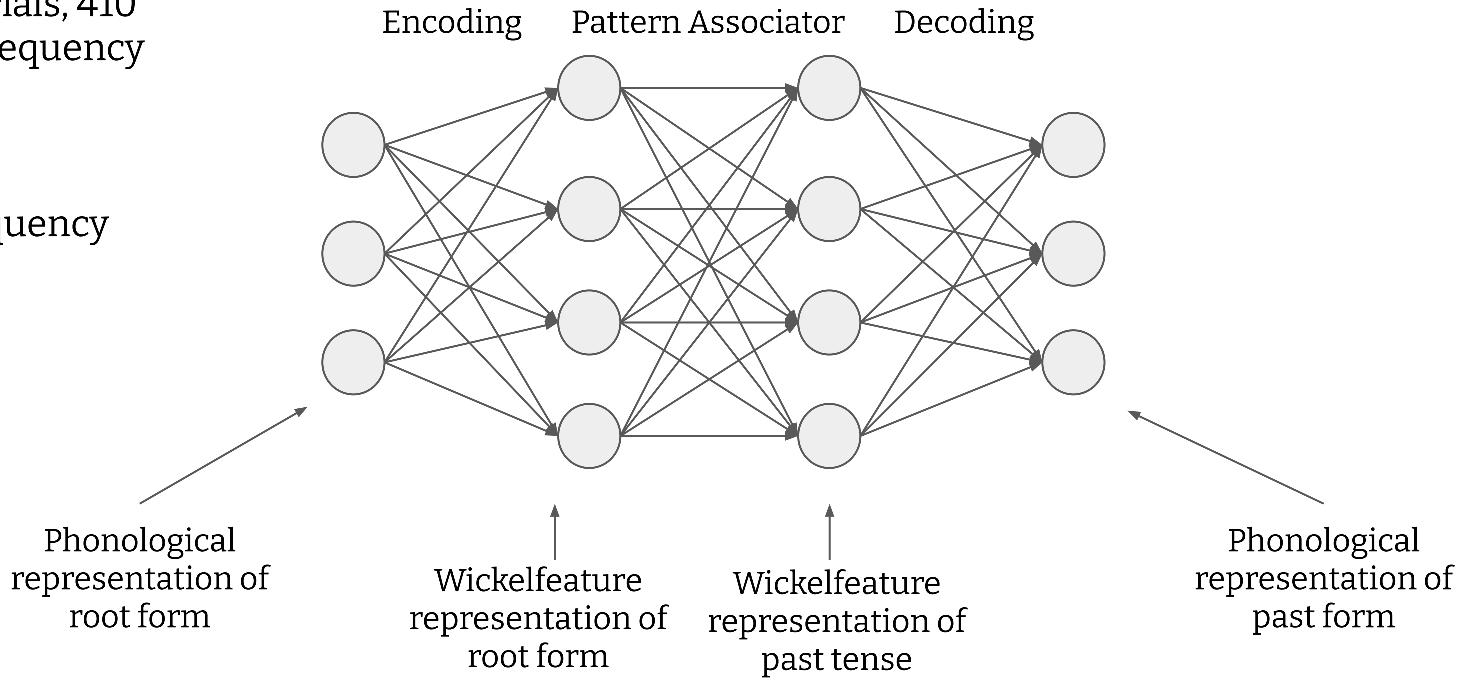


Figure adapted from paper

Connectionist model

root	eat
phonetic	/eet/
wickelphones of /eet/	${}_{*}e_{e}$ ${}_{e}e_{t}$ ${}_{e}t_{*}$
wickefeature of ${}_{*}e_{e}$	*: [(000) (00) (000) (00) 1] e: [(001) (01) (100) (01) 0] e: [(001) (01) (100) (01) 0]

Connectionist model

Pattern associators allow:

1. Exploitation of regularities that exist in mappings (e.g. dependent set of inputs -> patterns)
2. Regular patterns and exceptions to those patterns to coexist
3. For regularization, followed by the gradual tuning of connections to include exceptions

Discussion

1: Model captures U-learning three stage pattern

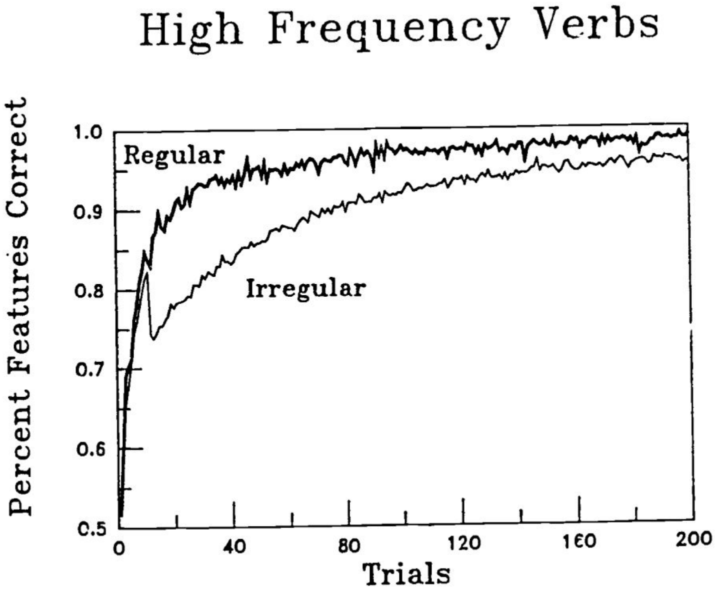


FIGURE 4. The percentage of correct features for regular and irregular high-frequency verbs as a function of trials.

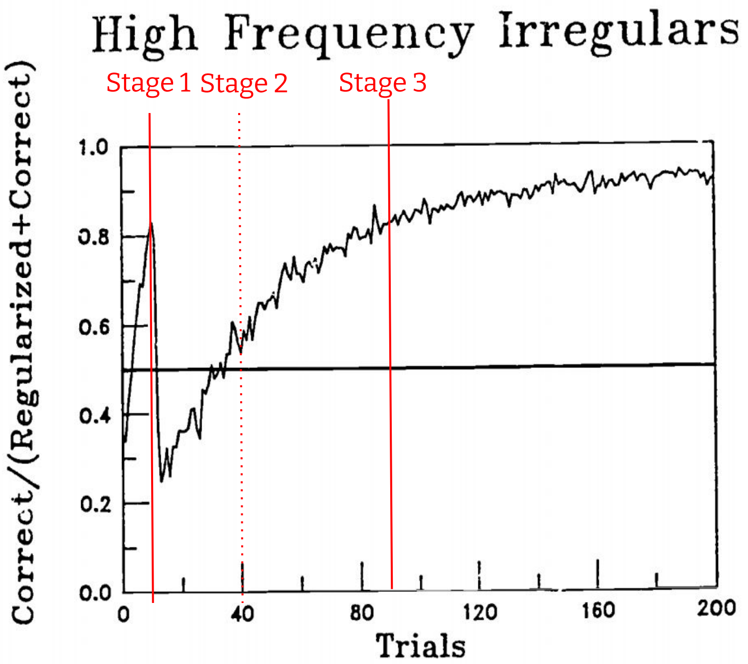


FIGURE 6. The ratio of the correct response to the sum of the correct and regularized response. Points on the curve below the .5 line are in the region where the regularized response is greater than the correct response.

Figures from paper

Discussion

2: Model captures differences in regular & irregular verbs

not t/d: drink, move, make
 -> used as no-change verbs
t/d: eat, build, pat
 -> predominantly regularized

TABLE 11

AVERAGE SIMULATED STRENGTHS OF REGULARIZED AND NO-CHANGE RESPONSES

Time Period	Verb Ending	Regularized	No Change
11-15	not *t/d*	0 44	0 10
	t/d	0 35	0 27
16-20	not *t/d*	0 32	0 12
	t/d	0 25	0 35
21-30	not *t/d*	0 52	0 11
	t/d	0.32	0 41

Table from paper

2: Model captures differences in regular & irregular verbs

TABLE 12

AVERAGE NUMBER OF WICKELFEATURES INCORRECTLY GENERATED

Trial	Irregular Verbs		Regular Verbs		
Number	Type I	Types III-VIII	Ending in *t/d*	Not Ending in *t/d*	CV*t/d*
11-15	89.8	123.9	74.1	82.8	87.3
16-20	57.6	93.7	45.3	51.2	60.5
21-30	45.5	78.2	32.9	37.4	47.9
31-50	34.4	61.3	22.9	26.0	37.3
51-100	18.8	39.0	11.4	12.9	21.5
101-200	11.8	21.5	6.4	7.4	12.7

no change　　vowel change

Table from paper

2: Model captures differences in regular & irregular verbs

TABLE 13

PERCENTAGE OF REGULARIZATION BY PRESCHOOLERS (Data from Bybee & Slobin, 1982)

Verb Type	Example	Percentage Regularizations
VIII	blew	80
VI	sang	55
V	bit	34
VII	broke	32
III	felt	13
IV	caught	10

TABLE 14

STRENGTH OF REGULARIZATION RESPONSES RELATIVE TO CORRECT RESPONSES

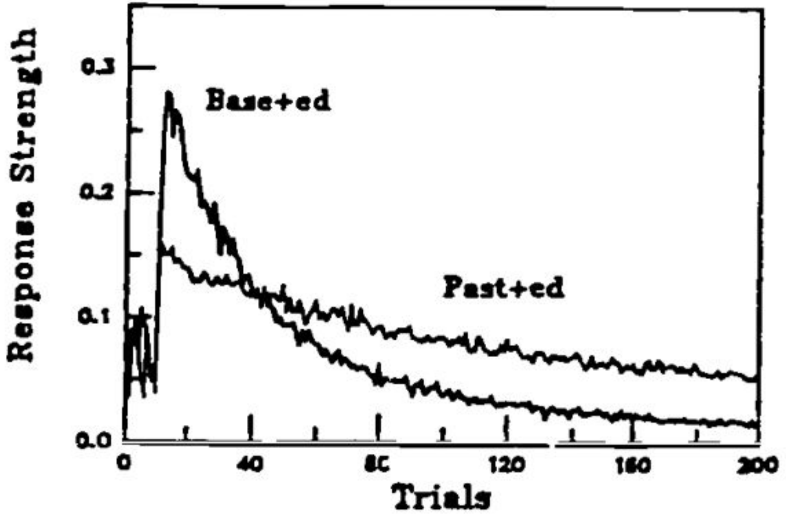
Rank Order	Data		Trials 11-15		Trials 16-20		Trials 21-30		Average Trials 11-30	
	Type	Percent	Type	Ratio	Type	Ratio	Type	Ratio	Type	Ratio
1	VIII	80	VIII	.86	VIII	.76	VIII	.61	VIII	.71
2	VI	55	VII	.80	VII	.74	VII	.61	VII	.69
3	V	34	VI	.76	V	.60	IV	.48	V	.56
4	VII	32	V	.72	IV	.59	V	.46	IV	.56
5	III	13	IV	.69	III	.57	III	.44	III	.53
6	IV	10	III	.67	VI	.52	VI	.40	VI	.52

Tables from paper

2: Model captures differences in regular & irregular verbs

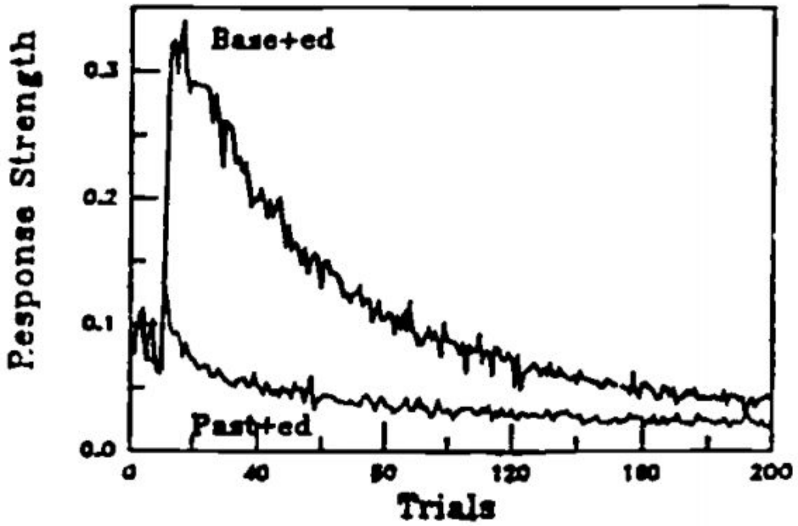
Verb Types II, V, VI, and VII

Examples: spend/spent; bite/bit; sing/sang; come/came



Verb Types III, IV, and VIII

sleep/slept; catch/caught; see/saw



Graphs from paper

Discussion

3: Model responds to training and testing sets

- The testing sample contains 86 “unseen" low frequency verbs (14 irregular and 72 regular), all of which were not chosen at random.
    - Six verbs had no response alternatives: jump, pump, soak, warm, trail, and glare
- 93% error rate for irregular verbs; 33% error rate for regular verbs
- 43% error rate overall

TABLE 17

THE MODEL'S RESPONSES TO UNFAMILIAR LOW-FREQUENCY IRREGULAR VERBS

Verb Type	Presented Word	Phonetic Input	Phonetic Response	English Rendition	Response Strength
I	bid	/bid/	/bid/	(bid)	0.55
	thrust	/Tr^st/	/Tr^st^d/	(thrusted)	0.57
II	bend	/bend/	/bend^d/	(bended)	0.28
	lend	/lend/	/lend^d/	(lended)	0.70
III	creep	/krEp/	/krEpt/	(creeped)	0.51
	weep	/wEp/	/wEpt/	(weeped)	0.34
			/wept/	(wept)	0.33
IV	catch	/kac/	/kact/	(catched)	0.67
V	breed	/brEd/	/brEd^d/	(breeded)	0.48
	grind	/grind/	/grind/	(grind)	0.44
	wind	/wind/	/wind/	(wind)	0.37
VI	cling	/kliN/	/kliNd/	(clinged)	0.28
			/kl^N/	(clung)	0.23
	dig	/dig/	/digd/	(digged)	0.22
	stick	/stik/	/stikt/	(sticked)	0.53
VII	tear	/ter/	/terd/	(teared)	0.90

TABLE 18

SYSTEM RESPONSES TO UNFAMILIAR LOW-FREQUENCY REGULAR VERBS

Verb Type	Presented Word	Phonetic Input	Phonetic Response	English Rendition	Response Rendition
End in t/d	guard	/gard/	/gard/	(guard)	0.29
			/gard^d/	(guarded)	0.26
	kid	/kid/	/kid/	(kid)	0.39
			/kid^d/	(kidded)	0.24
	mate	/mAt/	/mAt^d/	(mated)	0.43
			/mAd^d/	(maded)	0.23
	squat	/skw*t/	/skw*t^d/	(squated)	0.27
			/skw*t/	(squat)	0.22
			/skw*kt/	(squawked)	0.21
End in unvoiced consonant	carp	/karp/	/karpt/	(carped)	0.28
			/kapt^d/	(carpted)	0.21
	drip	/drip/	/dript^d/	(dripted)	0.28
			/dript/	(dripped)	0.22
	map	/map/	/mapt^d/	(mapted)	0.24
			/mapt/	(mapped)	0.22
	shape	/SAp/	/SApt/	(shaped)	0.43
			/Sipt/	(shipped)	0.27
	sip	/sip/	/sipt/	(sipped)	0.42
			/sept/	(sepped)	0.28
	slip	/slip/	/slept/	(slept)	0.40
	smoke	/smOk/	/smOkt^d/	(smokted)	0.29
			/smOk/	(smoke)	0.22
	snap	/snap/	/snapt^d/	(snapted)	0.40
	step	/step/	/stept^d/	(stepted)	0.59
	type	/tip/	/tipt^d/	(typted)	0.33
End in voiced consonant or vowel	brown	/brwn/	/brwnd/	(browned)	0.46
			/br*nd/	(brawned)	0.39
	hug	/h^g/	/h^g/	(hug)	0.59
	mail	/mA^l/	/mA^ld/	(mailed)	0.38
			/memb^ld/	(membled)	0.23
	tour	/tur/	/turd^r/	(toureder)	0.31
			/turd/	(toured)	0.25

Discussion

Compositional Generalization in Semantic Parsing: Pre-training vs. Specialized Architectures

Furrer, Zee, Scales, Schärli
Google Research

“How can we achieve compositional generalization in natural language?

1. How to properly measure compositional generalization?
2. Approaches tried
3. Which work? Which don’t? Future directions?

1. How to measure compositional generalization?

One way: The SCAN dataset

jump	⇒	JUMP
jump left	⇒	LTURN JUMP
jump around right	⇒	RTURN JUMP RTURN JUMP RTURN JUMP RTURN JUMP
turn left twice	⇒	LTURN LTURN
jump thrice	⇒	JUMP JUMP JUMP
jump opposite left and walk thrice	⇒	LTURN LTURN JUMP WALK WALK WALK
jump opposite left after walk around left	⇒	LTURN WALK LTURN WALK LTURN WALK LTURN WALK LTURN LTURN JUMP

Figure 1. Examples of SCAN commands (left) and the corresponding action sequences (right).

1. How to measure compositional generalization?

Template	Command	Target
1	"*turn left*"	LTURN
2	"*turn right*"	RTURN
3	"Primitive *left*"	LTURN [Primitive]
4	"Primitive *right*"	RTURN [Primitive]
5	"*turn opposite left*"	LTURN LTURN
6	"*turn opposite right*"	RTURN RTURN
7	"Primitive *opposite left*"	LTURN LTURN [Primitive]
8	"Primitive *opposite right*"	RTURN RTURN [Primitive]
9	"*turn around left*"	LTURN LTURN LTURN LTURN
10	"*turn around right*"	RTURN RTURN RTURN RTURN
11	"Primitive *around left*"	LTURN [Primitive] LTURN [Primitive] LTURN [Primitive] LTURN [Primitive]
12	"Primitive *around right*"	RTURN [Primitive] RTURN [Primitive] RTURN [Primitive] RTURN [Primitive]

Table 1: All command templates in the SCAN dataset, along with the target output. Here, "Primitive" can stand for "*jump*", "*walk*", "*run*", or "*look*", with the corresponding output [Primitive] being "JUMP", "WALK", "RUN", or "LOOK".

Traditional SCAN splits

Split name	Commands held out
Add jump	any compound containing "jump"
Add turn left	any compound containing "turn left"
Jump around right	any compound containing "jump around right"
Around right	any compound containing "PRIMITIVE around right" e.g. walk around right
Opposite right	any compound containing "PRIMITIVE opposite right"
Right	any compound containing "PRIMITIVE right"
Length	any command whose target sequence length is greater than 22

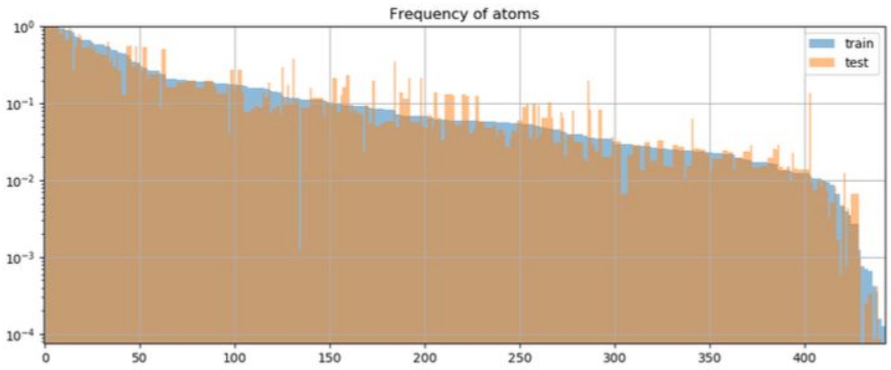
Distribution-Based Compositionality Assessment (DBCA) and Maximum Compound Divergence (MCD)

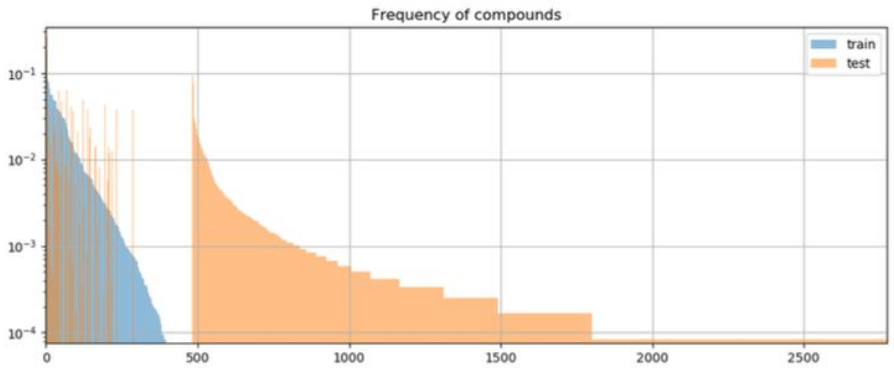
1. *Similar atom distribution*: All atoms present in the test set are also present in the train set, and the distribution of atoms in the train set is as *similar* as possible to their distribution in the test set.
2. *Different compound distribution*: The distribution of compounds in the train set is as *different* as possible from the distribution in the test set.

$$\mathcal{D}_C(V\|W) = 1 - C_{0.1}(\mathcal{F}_C(V) \,\|\, \mathcal{F}_C(W))$$
$$\mathcal{D}_A(V\|W) = 1 - C_{0.5}(\mathcal{F}_A(V) \,\|\, \mathcal{F}_A(W))$$

MCD: Split with maximum compound divergence $\mathcal{D}_C$, low atom divergence ($\mathcal{D}_A \leq 0.02$)

Distribution-Based Compositionality Assessment (DBCA) and Maximum Compound Divergence (MCD)





Frequency of atoms (left) and compounds (right) in the train and test sets of the MCD split for CFQ data

The CFQ Dataset

- Given natural language question, generate SPARQL query which, when executed, generates the correct answer

Table 1: Examples of generated questions at varying levels (L) of complexity.

L	Question $\mapsto$ Answer
10	What did [Commerzbank] acquire? $\mapsto$ Eurohypo; Dresdner Bank
15	Did [Dianna Rhodes]'s spouse produce [Soldier Blue]? $\mapsto$ No
20	Which costume designer of [E.T.] married [Mannequin]'s cinematographer? $\mapsto$ Deborah Lynn Scott
30	Who was influenced by and influenced [Steve Vai], [Marx Brothers], [Woody Allen], and [Steve Martin]? $\mapsto$ Brendon Small
40	Was [Weekend Cowgirls] produced, directed, and written by a film editor that [The Evergreen State College] and [Fairway Pictures] employed? $\mapsto$ No
50	Were [It's Not About the Shawerma], [The Fifth Wall], [Rick's Canoe], [White Stork Is Coming], and [Blues for the Avatar] executive produced, edited, directed, and written by a screenwriter's parent? $\mapsto$ Yes

2. Architectures and Techniques

- SCAN-inspired
  → Syn-att, CGPS, Equivariant, CNN, GECA
- Meta-learning
  → Meta seq2seq, Synth
- Symbolic
  → LANE
- MLM + Pretraining
  → T5 transformer family
- Other
  → NSEN

Results

Model	Add jump	Add turn left	Jump around right	Around right	Opposite right	Right	Length	SCAN MCD	CFQ MCD
LSTM	0.1	90.3	98.4 ±0.5	2.5 ±2.7	47.6 ±17.7	23.5 ±8.1	13.8	-	-
LSTM+A	0.0 ±0.0	82.6 ±8.2	**100.0 ±0.0**	0.0 ±0.0	16.5 ±6.4	30.0 ±7.8	14.1	6.1 ±1.7	14.9 ±1.1
CNN	69.2 ±9.2	-	-	56.7 ±10.2	-	-	0.0	-	-
GRU	12.5 ±6.6	59.1 ±16.8	-	-	-	-	18.1	-	-
GRU-dep	0.7 ±0.4	90.8 ±3.6	-	-	-	-	17.8	-	-
Transformer	1.0 ±0.6	**99.6 ±0.8**	**100.0 ±0.0**	53.3 ±10.9	3.0 ±6.8	92.0 ±15.1	0.0	0.9 ±0.3	17.8 ±0.9
Univ. Trans.	0.3 ±0.3	**99.4 ±1.4**	**100.0 ±0.0**	47.0 ±10.0	15.2 ±13.0	83.2 ±18.2	0.0	1.1 ±0.6	18.9 ±1.4
Evol. Trans.	0.6 ±0.6	**100.0 ±0.0**	**100.0 ±0.0**	30.2 ±28.4	11.6 ±14.6	**99.9 ±0.3**	19.8 ±0.0	1.6 ±0.6	20.8 ±0.7
Syn-att	91.0 ±27.4	**99.9 ±0.2**	98.9 ±2.3	28.9 ±34.8	10.5 ±8.8	99.1 ±1.8	15.2 ±0.7	-	-
CGPS	98.8 ±1.4	**99.7 ±0.4**	**100.0 ±0.0**	83.2 ±13.2	89.3 ±5.5	**99.7 ±0.5**	20.3 ±1.1	2.0 ±0.7	7.1 ±1.8
Equivariant*	99.1 ±0.0	-	-	92.0 ±0.2	-	-	15.9 ±3.2	-	-
GECA*	87.0 ±1.0	-	-	82.0 ±4.0	-	-	-	-	-
LANE	**100.0**	-	-	**100.0**	-	-	**100.0**	**100.0**	-
Meta seq2seq*	**99.9**	-	-	**99.9**	-	-	16.6	-	-
Synth*	**100.0**	-	-	**100.0**	-	-	**100.0**	-	-
NSEN	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	1.7 ±0.9	2.8 ±0.3
T5-small-NP	1.4 ±0.8	45.7 ±15.4	**100.0 ±0.0**	5.3 ±4.6	30.5 ±8.7	44.6 ±11.2	19.4 ±0.8	0.9 ±0.5	21.4 ±1.5
T5-small	84.1 ±1.0	73.0 ±5.8	**100.0 ±0.0**	31.8 ±1.0	58.2 ±10.4	88.7 ±8.9	10.9	6.9 ±1.1	28.0 ±0.6
T5-base	**99.5 ±0.0**	62.0 ±0.9	99.3 ±0.3	33.2 ±0.5	99.2 ±0.2	73.5 ±1.8	14.4	15.4 ±1.1	31.2 ±1.3
T5-large	98.3	69.2	**99.9**	46.8	**100.0**	91.0	5.2	10.1 ±1.6	34.8 ±1.5
T5-3B	99.0	65.1	**100.0**	27.4	90.0	76.6	3.3	11.6	40.2 ±4.2
T5-11B	98.3	87.9	**100.0**	49.2	99.1	91.1	2.0	9.1	40.9 ±4.3
T5-11B-mod	-	-	-	-	-	-	-	-	**42.1 ±9.1**

Pretraining success?

- Length split accuracy **decreases** as model size increases!
  19.4, 10.9, 14.4, 5.2, 3.3, 2.0
- SCAN MCD split accuracy with size shows no clear relation
  0.9, 6.0, 15.4, 10.1, 11.6, 9.1
- CFQ accuracy increases with size:
  21.5, 28.0, 31.2, 34.8, 40.2, 40.9
- Intermediate representation gives **+1.2%** accuracy boost

- Hypothesized benefit of pretraining: “improve model’s ability to substitute similar words by ensuring they are close to each other in representation space”
  - Achieves near-perfect performance on **Add jump** split, lesser gains on others.

Discussion

Symbolic approach: LANE

- Two modules, Composer and Solver, plus memory. Trained with curriculum and hierarchical RL.
- 100% accuracy on SCAN MCD split.

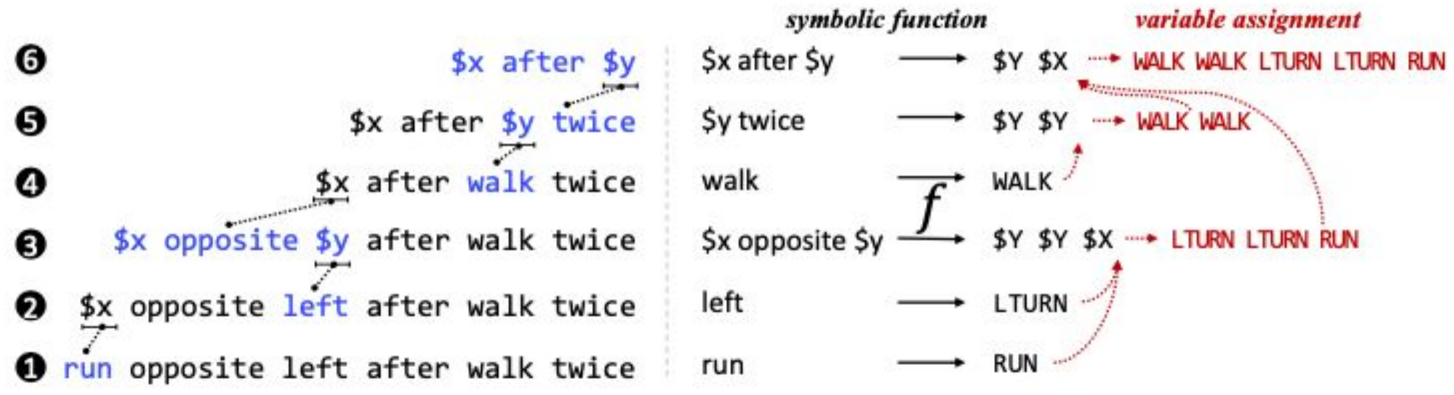
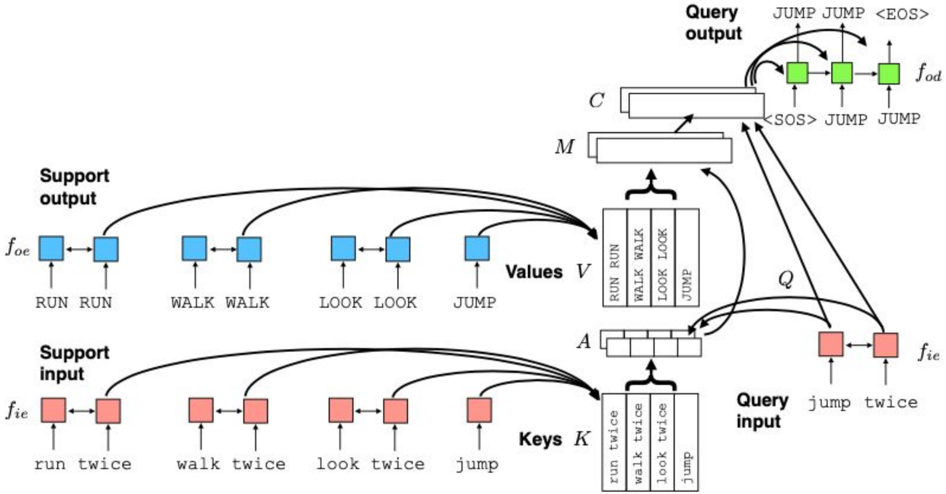


Figure 1: The schematic illustration of our idea on learning analytical expressions (see text).

Meta-learning: Meta seq2seq

- Trains over permutations of the SCAN grammar by remapping primitives to different outputs, e.g. jump -> WALK.
- Highly augmented training data — fair comparison?
- Builds invariance to primitive replacement in similar manner to Synth, Equivariant, and GECA approaches



Meta-learning: Synth

- seq2seq model takes in i/o examples and generates single program (interpretation grammar) which is symbolically evaluated to solve all examples.
- Trained by sampling grammars from a meta-grammar, and learning to output the correct program given examples generated with the sampled grammar.

```
G =
twice -> WALK
jump -> RTURN
turn -> JUMP
walk ->
blicket -> GREEN
kiki -> RUN
right -> RED
run -> BLUE
x2 left -> [x2] [x2] [x2] [x2] [x2]
x1 dax u1 -> [u1] [x1] [u1]
u1 thrice x2 -> [u1] [x2] [x2] [u1] [u1]
x1 look u2 -> [x1] [x1] [u2] [x1]
x2 around -> [x2]
u1 u2 -> [u2] [u1]
```

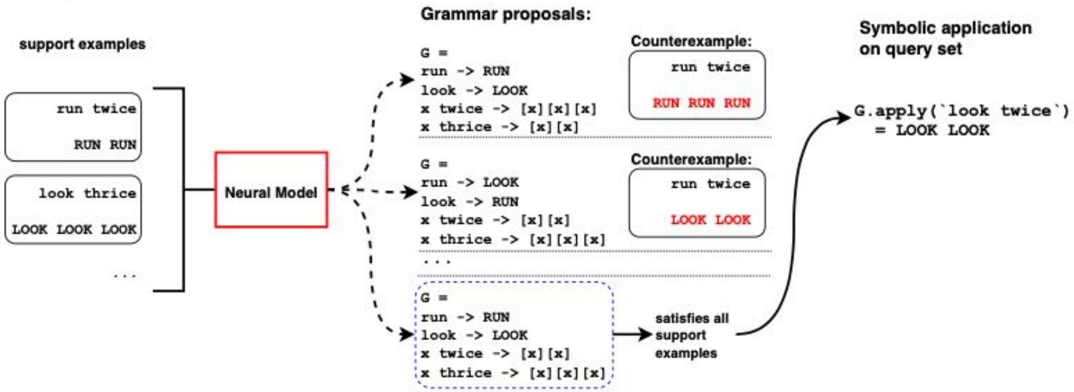
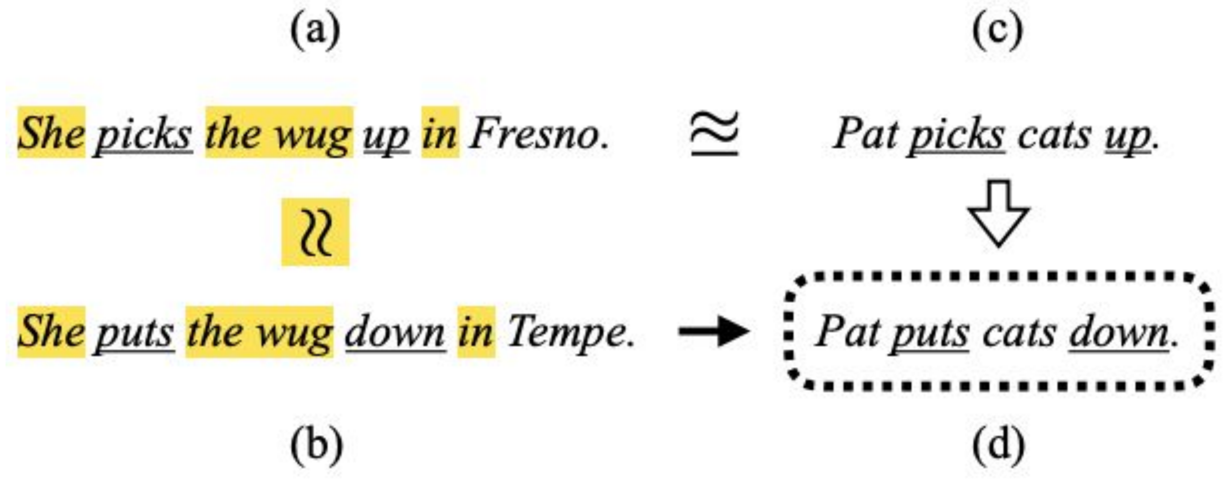


Figure 9. Samples from the training meta-grammar for SCAN.

GECA

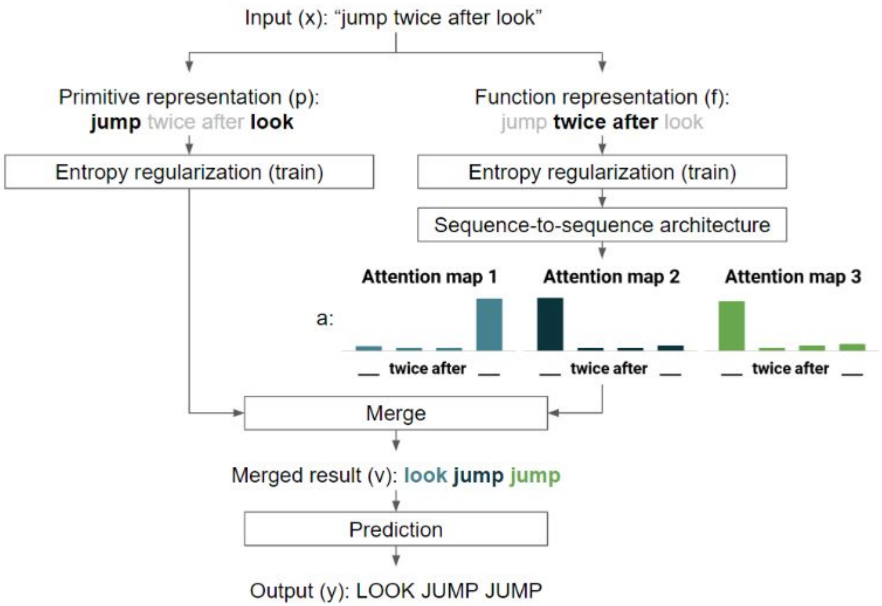
- Simple, effective approach: detects templates repeated during training, generates new training examples by filling with different fragments
- Augmenting training set so helps build invariance to compositional shifts in distribution



CGPS and Syn-att

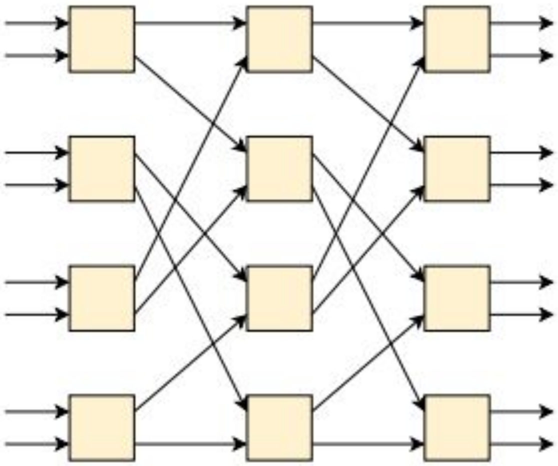
- Separates syntax (output action type) from semantics (output action order), each having a separate representation.
- CGPS chosen representative of SCAN-inspired approaches
- Bad performance on SCAN MCD

“It appears rather that the CGPS mechanism, unlike pre-training, is not robust to shifts in compound distribution and even introduces negative effects in such circumstances.”



NSEN

- Learns O(n log n) seq2seq algorithms with a shuffle-exchange architecture. Successor to Neural GPU.



Conclusions

1. Pretraining helps for compositional generalization, but does not solve it.
2. Specialized architectures often do not transfer to new compositional generalization benchmarks
3. Improvements in seq2seq architectures leads to corresponding incremental improvements in compositional settings
4. MCD likely measures compositional generalization more thoroughly than the traditional SCAN splits

Discussion