



Probing Neural Network Generalization using Default Patterns

Brandon Prickett^{*a} Tianyi Niu^{*b} Katya Pertsova^b

University of Massachusetts Amherst^a, University of North Carolina at Chapel Hill^b

SIGMORPHON 2025 - May 3, 2025 - Albuquerque, NM

Introduction: The Notorious **English past tense**

IF (“swing”, “wring”, “stick”, ...) **THEN:** [ɪ] → [ʌ]

ELSE IF (“think”, “buy”, “fight”, ...) **THEN:** ʊ → [ɔ]

ELSE IF (“dream”, “sleep”, “sweep”, ...) **THEN:** [i] → [ɛ] + t

...

ELSE: **add -ed/** (has the widest, most heterogeneous distribution)

What if ELSE case rarely occurs?

Introduction: Research questions

[RQ 1] Can neural-net models capture **minority defaults** that have been claimed to require **symbolic rules** (e.g., Pinker & Prince, 1994)

- ◆ YES for simple patterns (even for a one-node model like a perceptron/LR)
- ◆ **But: not as an across-the-board default rule, but a disjunctive pattern + class competition!**

[RQ 2] If so, how are they represented?

Introduction: Prior Work

- **Rumelhart and McClelland (1986)** first neural-net model of past-tense; criticized for its inability to appropriately handle default regular pattern (**Pinker and Prince, 1988**)
- **Hare et. al. (1995)** addressed the criticism and explored the notation of a less frequent “default” inflectional rule
- **Kirov and Cotterell (2018)**: RNN Encoder-Decoder (ED) to learn English verb inflection; criticized by **Corkery et al. (2019)** for inconsistency across multiple simulations
- **McCurdy, K., Goldwater, S., & Lopez, A. (2020); Beser (2021)** on German minority pattern, models learned pattern but does not generalize like human-speakers

Experiments: Artificial Grammar

Limitations of modeling real-language data (e.g., German and Arabic):

1. No agreement on whether German and Arabic plurals involve a minority default pattern.
2. No agreement (in case of German) on what the conditioning factors are.
3. Complex interaction across semantic, morphological, and phonological features

Artificial Grammar Learning as an alternative:

Artificial pattern, controlled stimuli and features

Experiments: Pattern based on English plural allomorphy

If a word ends is a sibilant (s, z, ʃ, ʒ, tʃ, dʒ)

—> suffix A ([+strid])

Else if a word ends in a [-voice] segment

—> suffix B (narrow default)

ELSE (voiced Cs and V)

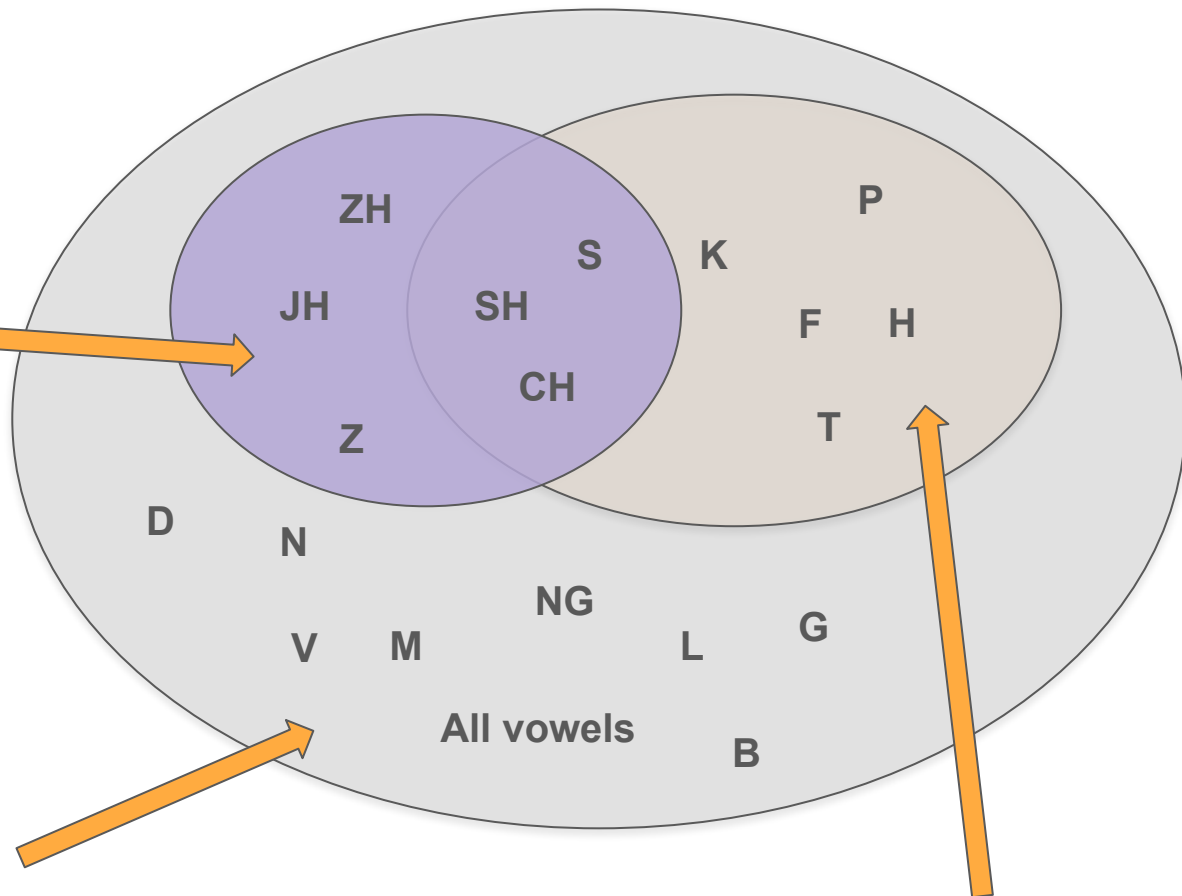
—> suffix C (wide default)

Majority Default	Equal Frequency	Minority Default
80% - suffix C 10% - suffix A 10% - suffix B	33% - suffix C 33% - suffix A 33% - suffix B	10% - suffix C 45% - suffix A 45% - suffix B

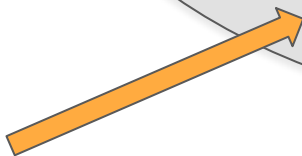
Experiments

Pattern visualized

Suffix A: [+strid]



Suffix C: everything else



Suffix B: [-voice] and not [+strid]



Experiments: Training Stimuli

Step 1: Randomly select template from {CVC, CVVC, VCVC, CVCV}

CVC

Step 2: Fill in segments

CVC → P IY1 H

Step 3: Determine suffix class

H ([-voice], [-strid]) → Suffix B

Step 4: Map input to feature vectors, output to one-hot vector

Input

	cons	syll	son	approx	voice	...
P	+	0	-	0	-	...
IY1	-	+	+	+	0	...
H	+	0	-	0	-	...

Output

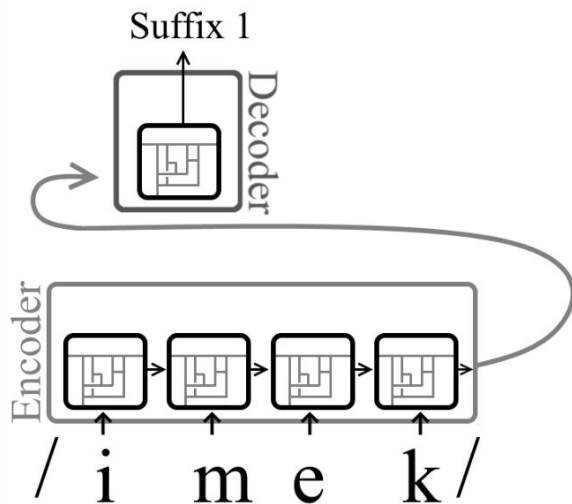
Suffix B → [0, 1, 0]

Experiments: Test Stimuli

1. **Held-out test set**
2. **Mutant words:** last segment changed to a different class → testing for effects of global similarity
3. **New segments ‘h’** (suffix B) and **‘l’** (suffix C) → testing feature-based generalization
4. **Novel templates:** {VC, CCV, CVCC} → testing “defaultness”
 - a. VC is the only 2-segment template, but is held-out

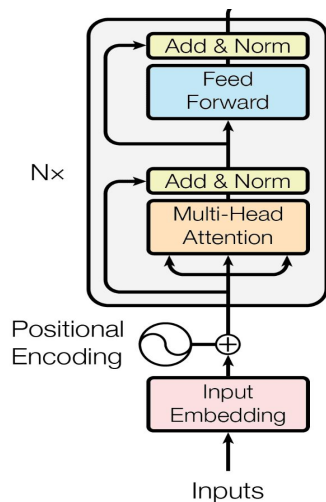
Experiments: Models

1. LSTM Encoder-decoder



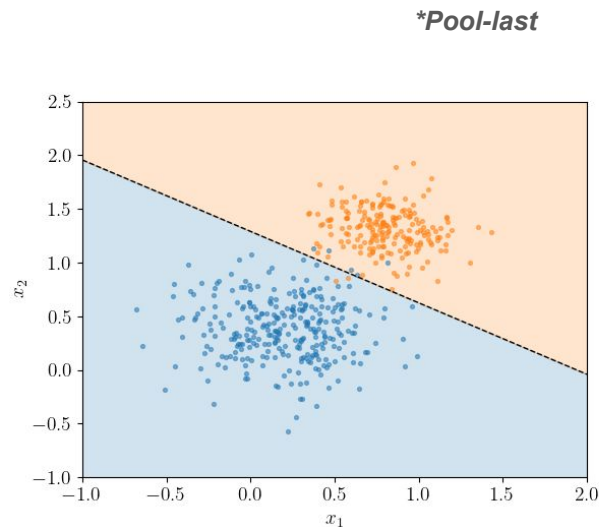
'Hard' temporal assumptions

2. Transformer Encoder



'Soft' temporal assumptions

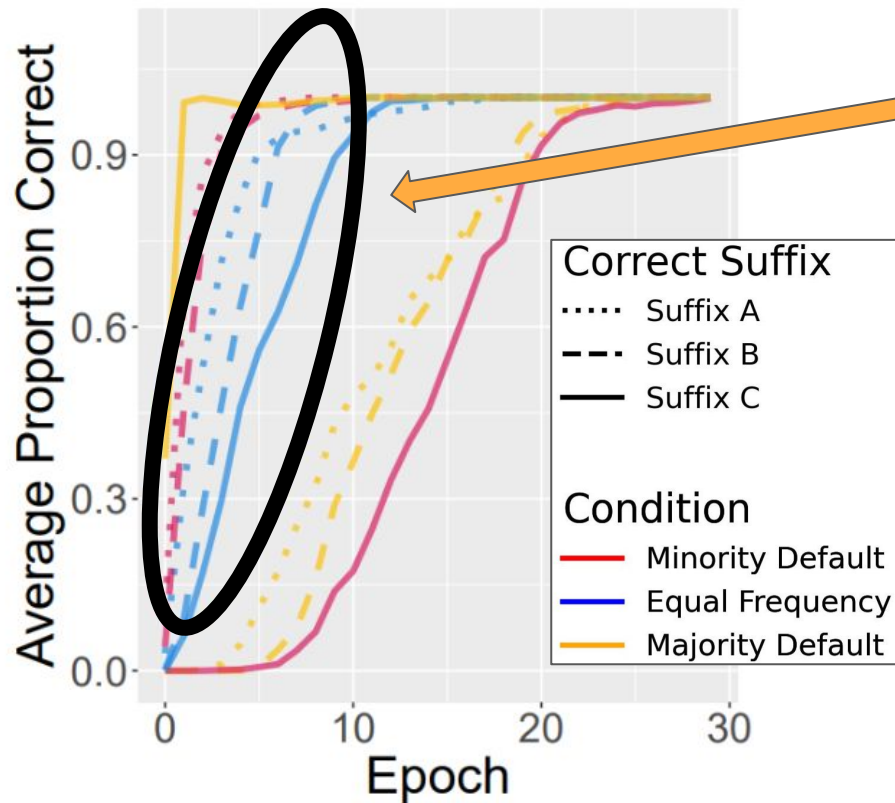
3. Logistic Regression



Linear decision boundary

Experiments: ED

Test Data learning curves

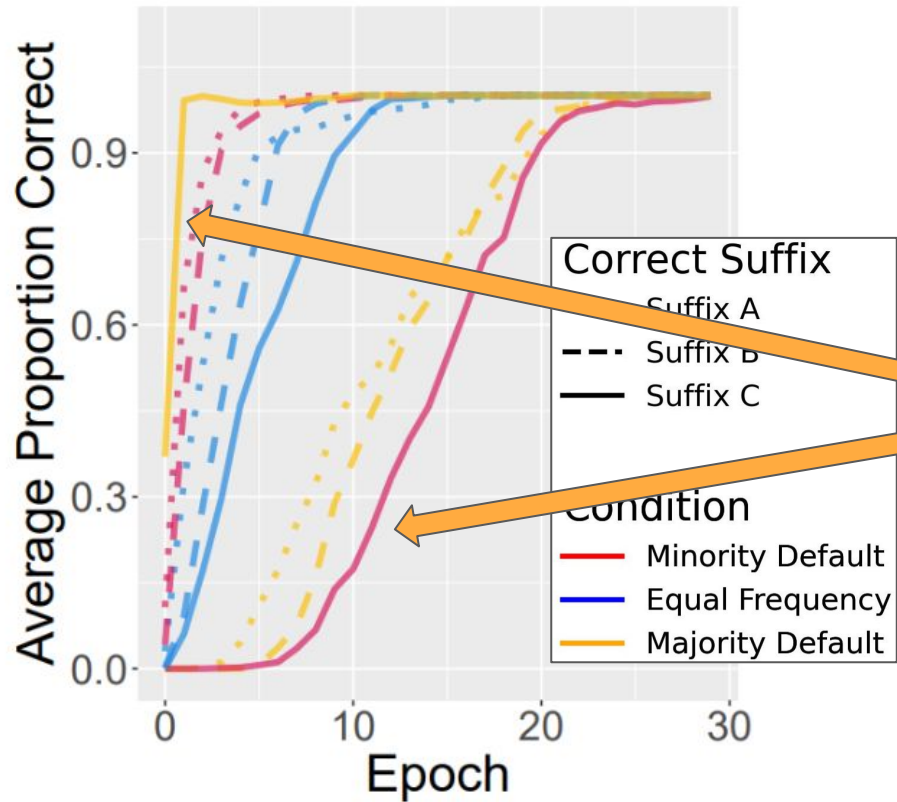


Distribution

Narrow distributions are learned first

Experiments: ED

Test Data learning curves



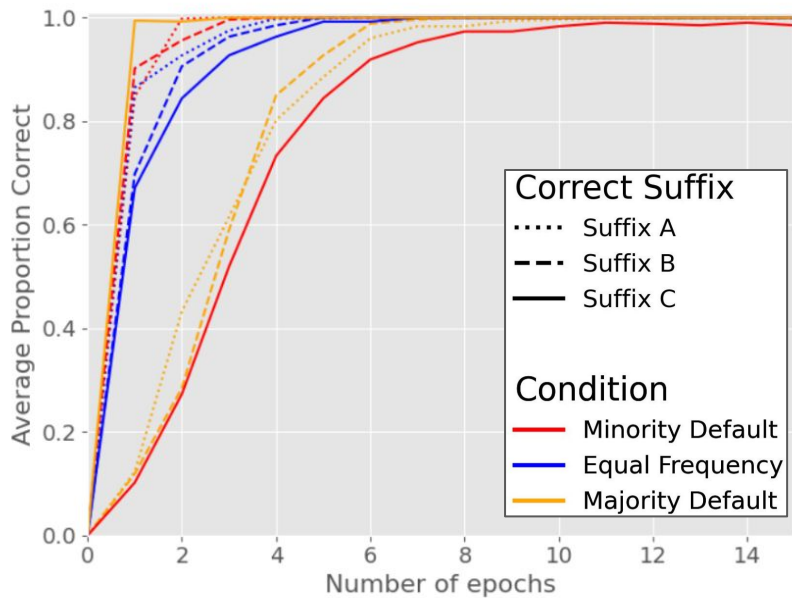
Distribution

Narrow distributions are learned first

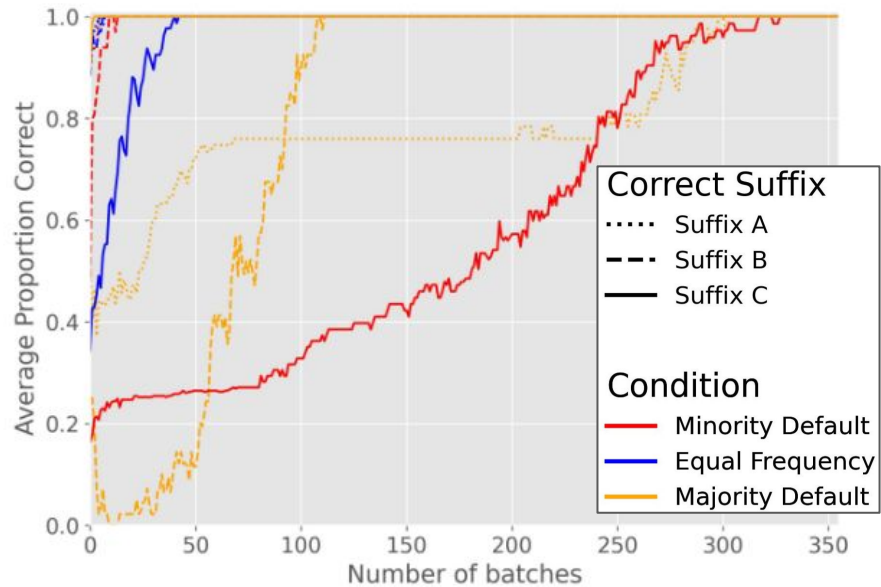
Frequency

More frequent patterns are learned first

Experiments: Transformer



Experiments: LR



Experiments: Accuracy on Test Stimuli

LSTM

Condition	Mutant	New Template	[h]-final	[l]-final
<i>Maj. Def.</i>	0.97	0.86	0.96	1.00
<i>Equal Freq.</i>	0.98	0.81	1.00	0.99
<i>Min. Def.</i>	1.00	0.83	1.00	0.89

Transformer

Mutant	New Template	[h]-final	[l]-final
1.00	0.82	0.73	0.96
1.00	0.76	1.00	0.97
1.00	0.83	1.00	0.84

Overall high performance on mutants

Experiments: Accuracy on Test Stimuli

LSTM

Condition	Mutant	New Template	[h]-final	[l]-final
<i>Maj. Def.</i>	0.97	0.86	0.96	1.00
<i>Equal Freq.</i>	0.98	0.81	1.00	0.99
<i>Min. Def.</i>	1.00	0.83	1.00	0.89

Transformer

Mutant	New Template	[h]-final	[l]-final
1.00	0.82	0.73	0.96
1.00	0.76	1.00	0.97
1.00	0.83	1.00	0.84

LSTM struggled with CVCC

- Potentially due to how the model finds the last consonant in a word

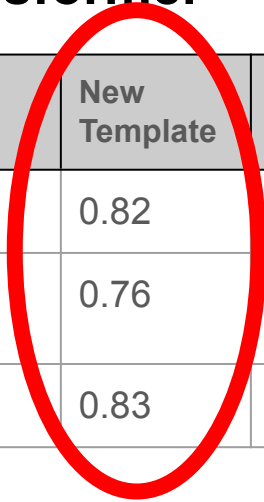
Experiments: Accuracy on Test Stimuli

LSTM

Condition	Mutant	New Template	[h]-final	[l]-final
<i>Maj. Def.</i>	0.97	0.86	0.96	1.00
<i>Equal Freq.</i>	0.98	0.81	1.00	0.99
<i>Min. Def.</i>	1.00	0.83	1.00	0.89

Transformer

Mutant	New Template	[h]-final	[l]-final
1.00	0.82	0.73	0.96
1.00	0.76	1.00	0.97
1.00	0.83	1.00	0.84



Transformer struggled with VC

- *But model generalized to suffix C. Evidence of default behavior?*

Experiments: Accuracy on Test Stimuli

RNN

Condition	Mutant	New Template	[h]-final	[l]-final
<i>Maj. Def.</i>	0.97	0.86	0.96	1.00
<i>Equal Freq.</i>	0.98	0.81	1.00	0.99
<i>Min. Def.</i>	1.00	0.83	1.00	0.89

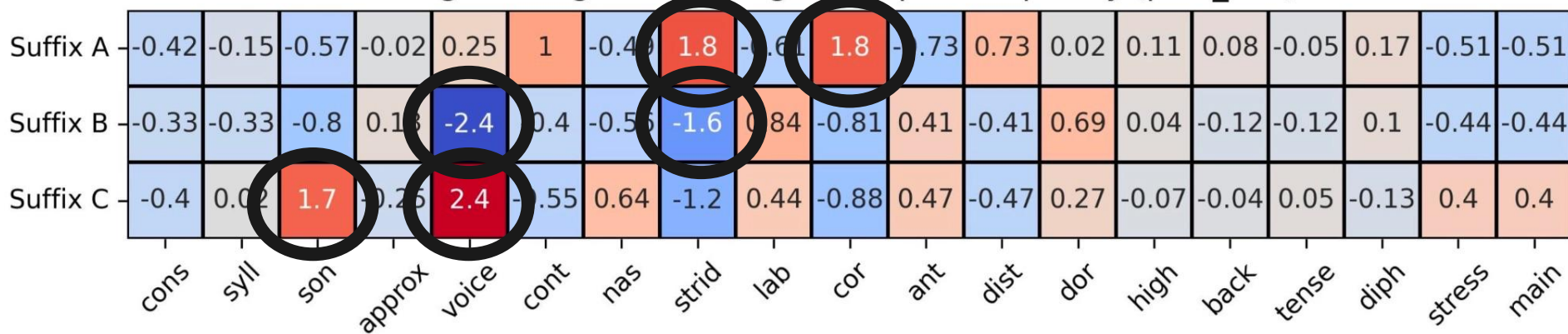
Transformer

Mutant	New Template	[h]-final	[l]-final
1.00	0.82	0.73	0.96
1.00	0.76	1.00	0.97
1.00	0.83	1.00	0.84

Better performance on [h] than [l]

Analysis: LR Heatmap

Logistic Regression Weights - Equal Frequency (pool_last)



Ground truth:

- Suffix A: [+strid]
- Suffix B: [-voice]
- Suffix C: everything else

Model learned:

- Suffix A: [+strid], [+cor]
- Suffix B: [-voice], [-strid]
- Suffix C: [+voice], [+son]

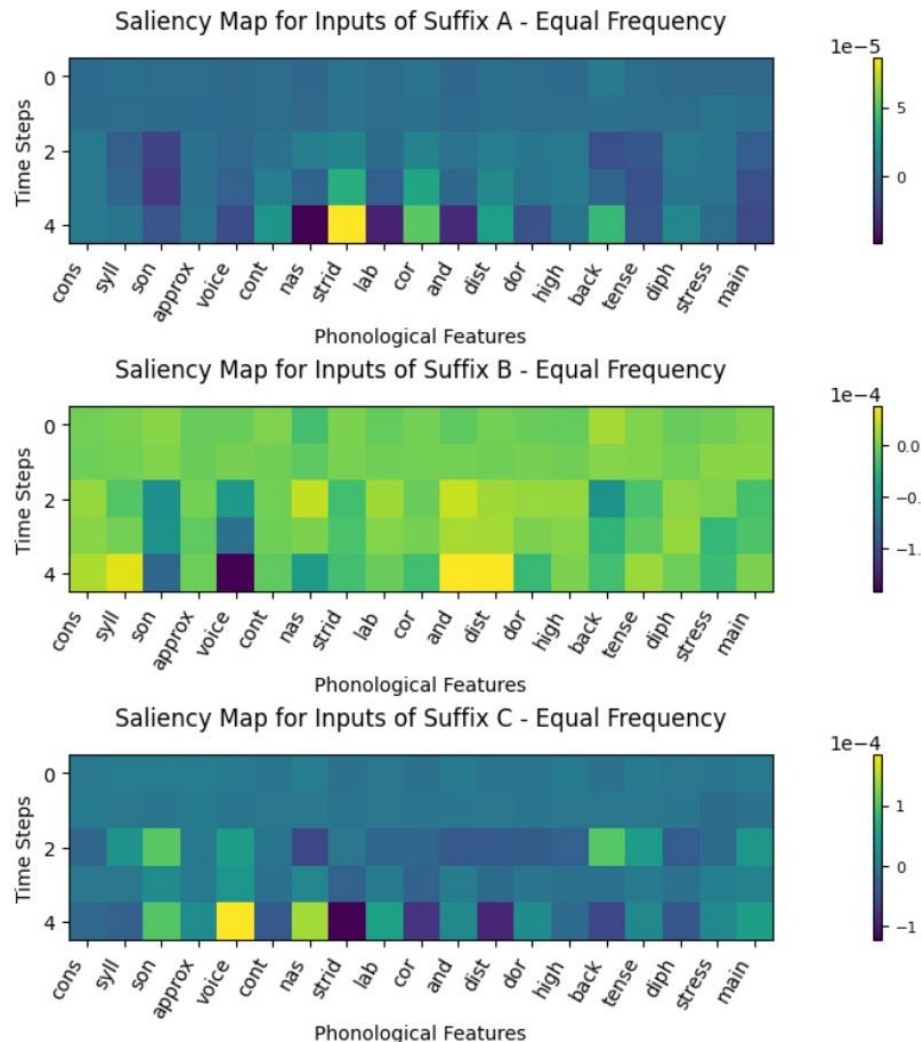
Analysis: Saliency Map

Saliency map: 'How sensitive is an output class to a specific feature?'

Yellow ~ positive; Blue ~ negative

Model learned:

- Suffix A: [+strid], [-nas]
- Suffix B: [-voice], [-son]
- Suffix C: [+voice], [-strid]



Key conclusions

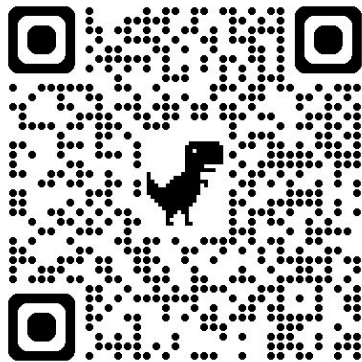
- All models learned the minority default pattern through class-competition
 - Effect of **distribution**: suffixes that apply to a more narrow class are learned better/first
 - Effect of **frequency**: frequent patterns are learned better/first
- Generalization to “novel” stimuli:
 - Mutants: models pick up the relevant last segment, unaffected by overall similarity
 - Novel templates - Transformers fails on VC & overgeneralizes the default

Future Directions

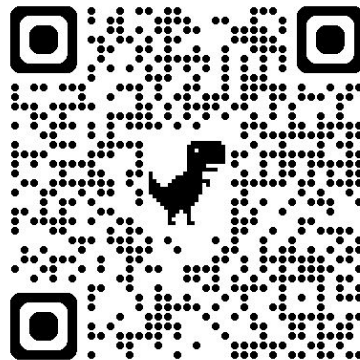
- More complex patterns (e.g., exceptions that don't form a natural class)
- Type vs. token frequency differences
- Different output representations

Thank you!

Code



Paper



Agenda

1. Background & Research questions
 - a. Can neural nets represent defaults, and minority defaults?
 - b. Previous findings
2. Experiments
 - a. Design
 - b. Models
 - c. Results
3. Takeaways and Future directions

Introduction

Big picture question

- Can neural-net models capture **minority defaults** that have been claimed to require **symbolic rules** (e.g., Pinker & Prince, 1994)
 - ◆ YES for simple patterns (even for a one-node model like a perceptron)
 - ◆ **But: not as an across-the-board default rule, but a disjunctive pattern+competition!**
- If so, how are they represented?

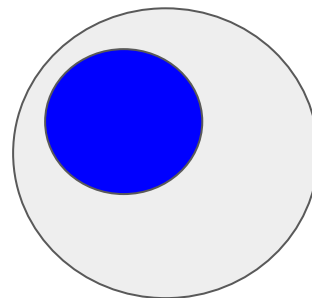
Default or Elsewhere Case:

IF (some condition holds):

Do X

ELSE:

Do Y → default



Introduction: Prior Work

- **Rumelhart and McClelland (1986)** first neural-net model of past-tense; criticized for its inability to appropriately handle default regular pattern (**Pinker and Prince, 1988**)
- **Hare et. al. (1995)** addressed the criticism and explored the notation of a less frequent “default” inflectional rule
- **Kirov and Cotterell (2018)**: RNN Encoder-Decoder (ED) to learn English verb inflection; criticized by **Corkery et al. (2019)** for inconsistency across multiple simulations
- **McCurdy, K., Goldwater, S., & Lopez, A. (2020); Beser (2021)** on German minority pattern, models learns frequency not defaults

Introduction

How do we know something is a default?

- Often, the test for whether a pattern is a “default” involves testing whether native speakers would generalize it to novel situations that don’t fit any other learned pattern.
- - E.g. Pinker’s Dual-Route Model (Pinker, 1991) of English Past Tense predicts **the regular -ed to occur by default** whenever associative memory fails or has no relevant words/patterns. That is, in cases of:
 - Borrowed or novel words, including phonotactically-odd or illegal ones
 - Abbreviations or acronyms
 - Words used with additional morphological structure (proper names, derivatives,)
 - In cases of certain types of aphasia or language disorders

Introduction

Introduction: Prior Work

- An early connectionist model by Rumelhart and McClelland (1986) was criticized by proponents of rule-based approaches for its inability to appropriately handle default regular pattern (Pinker and Prince, 1988)
 - The model's ability to handle the regular rule was attributed to the **frequency of -ed**. So, **minority default patterns** became a particular focus.
- Hare et. al. (1995) show a simple Feed-Forward network model can handle minority defaults (using pseudo-early English minority default pattern)
 - Pinker and Ullman (2002) *“clean-up network’ in which the units for -ed strengthen the units for an unchanged stem vowel and inhibit the units for a changed vowel [46] – in effect, an innate mechanism dedicated to the English past tense.”*

Introduction: More recent work

- Kirov and Cotterell (2018) propose an RNN deep learning model with the Encoder-Decoder (ED) architecture to learn patterns like the English past tense.
 - BUT: Corkery et al. (2019) showed that the ED model's predictions **were inconsistent across multiple simulations** and, when averaged together, did not closely match the human data (they were worse than the rule-based model)
- McCurdy, K., Goldwater, S., & Lopez, A. (2020). **Inflecting when there's no majority: limitations of encoder-decoder neural networks as cognitive models for German plurals.**
 - The model achieved 88.8% accuracy on held-out data, but failed to predict the pattern of behavior on unusual and novel-sounding words
- Beser (2021) compared a **transformer** model with an **RNN's** on German plural (also English past tense and Russian paradigms gaps)
 - Both models perform relatively well when given gender information
 - Both still fail to match human behavior on unusual stimuli from McCurdy et. Al. (2020)

Case Study

Logistic Regression Weights - Minority Default (pool_last)

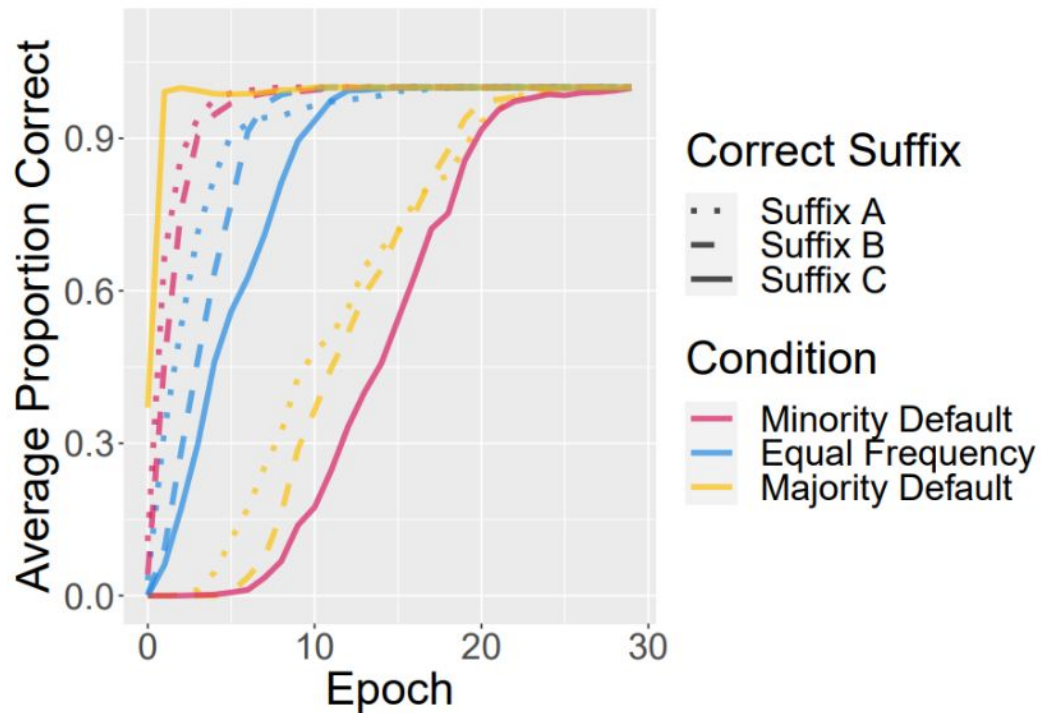
Suffix A	-0.35	-0.14	-0.16	-0.06	0.63	0.93	-0.26	1.9	-0.71	1.8	-0.69	0.69	-0.19	0.08	0.09	0.03	0.14	-0.32	-0.32
Suffix B	-0.17	-0.2	-0.57	0.13	-2	-0.6	-0.37	-1.8	0.89	-1.1	0.5	-0.5	0.56	0.06	-0.05	-0.06	0.05	-0.24	-0.24
Suffix C	-0.57	0.01	1.4	-0.21	1.8	-0.22	0.41	-0.82	0.46	-0.67	0.33	-0.33	0.46	-0.05	-0.05	-0.01	-0.09	0.21	0.21
	cons	syll	son	approx	voice	cont	nas	strid	lab	cor	ant	dist	dor	high	back	tense	diph	stress	main

Why does LR fail [l] only in minority default?

1. [l] underspecified or [voice] and [strid]
2. [cor] in Suffix A moves faster than [son] in Suffix C in minority default
3. [l] gets classified into Suffix A
4. [n] also underspecified, but in training set!

Experiments: ED

Test Data learning curves



ENLARGED ON NEXT SLIDE

Average Proportion Correct

0.9
0.6
0.3
0.0

0

10

20

30

Epoch

Correct Suffix

· · Suffix A
- Suffix B
— Suffix C

Condition

Minority Default
Equal Frequency
Majority Default

C is slowest
when freq. is
equal

Min. Def. learned
slowest

