Learning biases in opaque interactions

BRANDON PRICKETT

UNIVERSITY OF MASSACHUSETTS, AMHERST

NECPHON 2017

Overview

1. Background
    1. Phonological interactions
    2. Opaque interactions
    3. Kiparskian biases
    4. Recent approaches to interaction learning
2. Experimental data
    1. Methods
    2. Results
3. Comparing the models' performance
    1. Expectation Driven Learning with SMR constraints
    2. MaxEnt Stratal OT learner
    3. A Sequence-to-Sequence neural net
4. Conclusions
    1. Which learner predicted the data best?
    2. Future work

1. Background

Phonological interactions

- Phonological processes can interact with one another.
- Sometimes these interactions are transparent in the surface form.

Phonological interactions

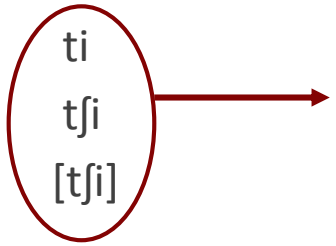
- Phonological processes can interact with one another.
- Sometimes these interactions are transparent in the surface form.
- Example of a toy ***Feeding*** interaction (all examples adapted from Baković 2011 and Jarosz 2016):

• UR	/ti/	/kai/	/tai/
• Vowel Deletion: V → Ø/_V	--	ki	ti
• Palatalization: T→[+palatal]/_[+front]	tʃi	--	tʃi
• SR	[tʃi]	[ki]	[tʃi]

Phonological interactions

- Phonological processes can interact with one another.
- Sometimes these interactions are transparent in the surface form.
- Example of a toy ***Feeding*** interaction (all examples adapted from Baković 2011 and Jarosz 2016):

	/ti/	/kai/	/tai/
UR	/ti/	/kai/	/tai/
Vowel Deletion: V → Ø/_V	--	ki	ti
Palatalization: T→[+palatal]/_[+front]	tʃi	--	tʃi
SR	[tʃi]	[ki]	[tʃi]



Deletion moves the /i/ so that it can trigger the /t/→ [tʃ] process.

Phonological interactions

- Phonological processes can interact with one another.
- Sometimes these interactions are transparent in the surface form.
- Example of a toy ***Feeding*** interaction (all examples adapted from Baković 2011 and Jarosz 2016):

• UR	/ti/	/kai/	/tai/
• Vowel Deletion: V → Ø/_V	--	ki	ti
• Palatalization: T→[+palatal]/_[+front]	tʃi	--	tʃi
• SR	[tʃi]	[ki]	[tʃi]

- Example of a toy ***Bleeding*** interaction:

• UR	/ti/	/kia/	/tia/
• Vowel Deletion: V → Ø/_V	--	ka	ta
• Palatalization: T→[+palatal]/_[+front]	tʃi	--	--
• SR	[tʃi]	[ka]	[ta]

Phonological interactions

- Phonological processes can interact with one another.
- Sometimes these interactions are transparent in the surface form.
- Example of a toy ***Feeding*** interaction (all examples adapted from Baković 2011 and Jarosz 2016):

• UR	/ti/	/kai/	/tai/
• Vowel Deletion: V → Ø/_V	--	ki	ti
• Palatalization: T→[+palatal]/_[+front]	tʃi	--	tʃi
• SR	[tʃi]	[ki]	[tʃi]

- Example of a toy ***Bleeding*** interaction:

• UR	/ti/	/kia/	/tia/
• Vowel Deletion: V → Ø/_V	--	ka	ta
• Palatalization: T→[+palatal]/_[+front]	tʃi	--	--
• SR	[tʃi]	[ka]	[ta]

Deletion removes the /i/ so that it can't trigger the /t/→ [tʃ] process.

Opaque interactions

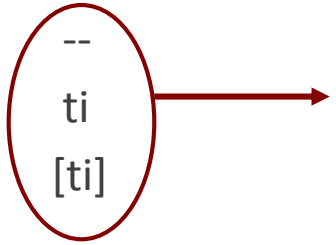
- However, these interactions can also be opaque (Kiparsky 1968, 1971).
- This can take the form of a process not applying when it seems like it should (counterfeeding) or applying when it seems like it shouldn't (counterbleeding).
- Example of a toy ***Counterfeeding*** interaction (adapted from Baković 2011 and Jarosz 2016):

• UR	/ti/	/kai/	/tai/
• Palatalization: T→[+palatal]/_[+front]	tʃi	--	--
• Vowel Deletion: V → Ø/_V	--	ki	ti
• SR	[tʃi]	[ki]	[ti]

Opaque interactions

- However, these interactions can also be opaque (Kiparsky 1968, 1971).
- This can take the form of a process not applying when it seems like it should (counterfeeding) or applying when it seems like it shouldn't (counterbleeding).
- Example of a toy ***Counterfeeding*** interaction (adapted from Baković 2011 and Jarosz 2016):

	/ti/	/kai/	/tai/
UR	/ti/	/kai/	/tai/
Palatalization: T→[+palatal]/_[+front]	tʃi	--	--
Vowel Deletion: V → Ø/_V	--	ki	ti
SR	[tʃi]	[ki]	[ti]



Deletion moves the /i/ next to the /t/, but does so too late to cause the /t/→[tʃ] process.

Opaque interactions

- However, these interactions can also be opaque (Kiparsky 1968, 1971).
- This can take the form of a process not applying when it seems like it should (counterfeeding) or applying when it seems like it shouldn't (counterbleeding).
- Example of a toy ***Counterfeeding*** interaction (adapted from Baković 2011 and Jarosz 2016):

• UR	/ti/	/kai/	/tai/
• Palatalization: T→[+palatal]/_[+front]	tʃi	--	--
• Vowel Deletion: V → Ø/_V	--	ki	ti
• SR	[tʃi]	[ki]	[ti]

- Example of a toy ***Counterbleeding*** interaction (adapted from Baković 2011 and Jarosz 2016):

• UR	/ti/	/kia/	/tia/
• Palatalization: T→[+palatal]/_[+front]	tʃi	--	tʃia
• Vowel Deletion: V → Ø/_V	--	ka	tʃa
• SR	[tʃi]	[ka]	[tʃa]

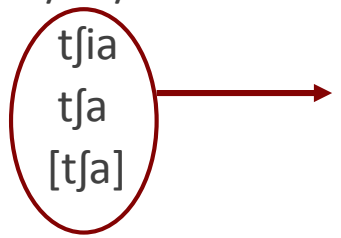
Opaque interactions

- However, these interactions can also be opaque (Kiparsky 1968, 1971).
- This can take the form of a process not applying when it seems like it should (counterfeeding) or applying when it seems like it shouldn't (counterbleeding).
- Example of a toy ***Counterfeeding*** interaction (adapted from Baković 2011 and Jarosz 2016):

UR	/ti/	/kai/	/tai/
Palatalization: T→[+palatal]/_[+front]	tʃi	--	--
Vowel Deletion: V → Ø/_V	--	ki	ti
SR	[tʃi]	[ki]	[ti]

- Example of a toy ***Counterbleeding*** interaction (adapted from Baković 2011 and Jarosz 2016):

UR	/ti/	/kia/	/tia/
Palatalization: T→[+palatal]/_[+front]	tʃi	--	tʃia
Vowel Deletion: V → Ø/_V	--	ka	tʃa
SR	[tʃi]	[ka]	[tʃa]



Deletion removes the /i/, but does so too late to stop the /t/→[tʃ] process.

Kiparskian biases

- When discussing phonological interactions, Kiparsky (1968, 1971) discusses two different learning biases that could influence the likelihood of these interactions arising diachronically:
  - Application bias (Kiparsky 1968) – this favors interactions that cause more processes to apply. Out of the interactions we just discussed, this bias favors feeding and counterbleeding.
  - Transparency bias (Kiparsky 1971) – this favors interactions that apply processes in an order that is transparent from the surface form (i.e. there are no surface violations of rules that applied somewhere in the grammar). Out of the interactions we just discussed, this bias favors feeding and bleeding.
- **Kiparsky didn't propose that *both* biases were at work, although I'll be testing that possibility.**

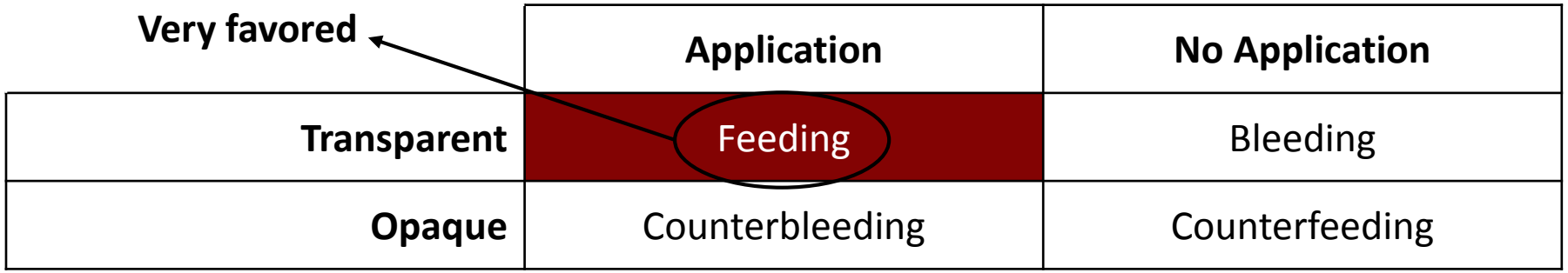
Kiparskian biases

- When discussing phonological interactions, Kiparsky (1968, 1971) discusses two different learning biases that could influence the likelihood of these interactions arising diachronically:
  - Application bias (Kiparsky 1968) – this favors interactions that cause more processes to apply. Out of the interactions we just discussed, this bias favors feeding and counterbleeding.
  - Transparency bias (Kiparsky 1971) – this favors interactions that apply processes in an order that is transparent from the surface form (i.e. there are no surface violations of rules that applied somewhere in the grammar). Out of the interactions we just discussed, this bias favors feeding and bleeding.

	Application	**No Application**
Transparent	Feeding	Bleeding
Opaque	Counterbleeding	Counterfeeding

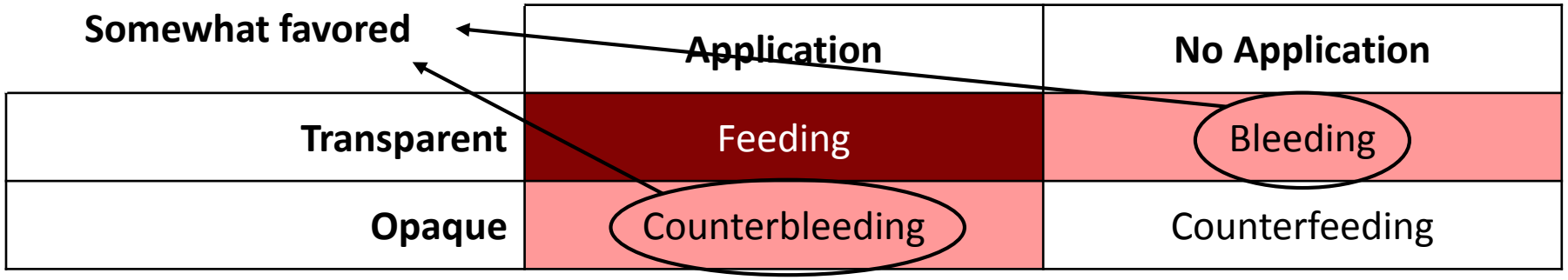
Kiparskian biases

- When discussing phonological interactions, Kiparsky (1968, 1971) discusses two different learning biases that could influence the likelihood of these interactions arising diachronically:
  - Application bias (Kiparsky 1968) – this favors interactions that cause more processes to apply. Out of the interactions we just discussed, this bias favors feeding and counterbleeding.
  - Transparency bias (Kiparsky 1971) – this favors interactions that apply processes in an order that is transparent from the surface form (i.e. there are no surface violations of rules that applied somewhere in the grammar). Out of the interactions we just discussed, this bias favors feeding and bleeding.



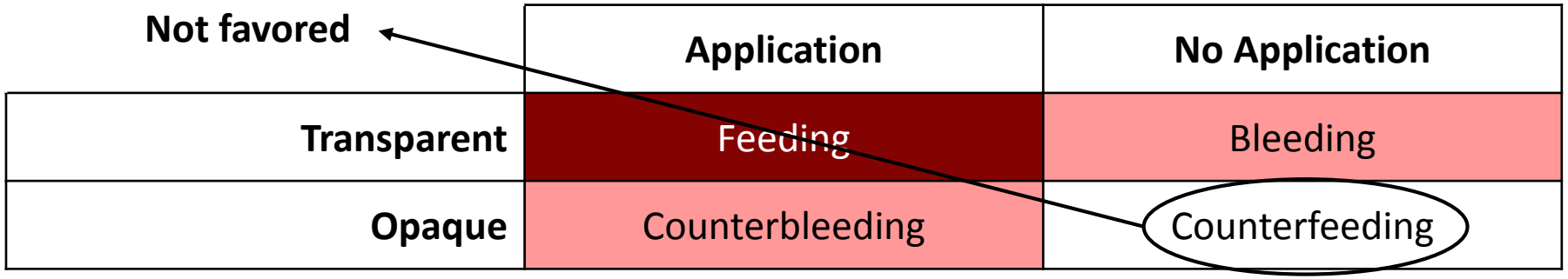
Kiparskian biases

- When discussing phonological interactions, Kiparsky (1968, 1971) discusses two different learning biases that could influence the likelihood of these interactions arising diachronically:
  - Application bias (Kiparsky 1968) – this favors interactions that cause more processes to apply. Out of the interactions we just discussed, this bias favors feeding and counterbleeding.
  - Transparency bias (Kiparsky 1971) – this favors interactions that apply processes in an order that is transparent from the surface form (i.e. there are no surface violations of rules that applied somewhere in the grammar). Out of the interactions we just discussed, this bias favors feeding and bleeding.



Kiparskian biases

- When discussing phonological interactions, Kiparsky (1968, 1971) discusses two different learning biases that could influence the likelihood of these interactions arising diachronically:
  - Application bias (Kiparsky 1968) – this favors interactions that cause more processes to apply. Out of the interactions we just discussed, this bias favors feeding and counterbleeding.
  - Transparency bias (Kiparsky 1971) – this favors interactions that apply processes in an order that is transparent from the surface form (i.e. there are no surface violations of rules that applied somewhere in the grammar). Out of the interactions we just discussed, this bias favors feeding and bleeding.



Kiparskian biases

- When discussing phonological interactions, Kiparsky (1968, 1971) discusses two different learning biases that could influence the likelihood of these interactions arising diachronically:
  - Application bias (Kiparsky 1968) – this favors interactions that cause more processes to apply. Out of the interactions we just discussed, this bias favors feeding and counterbleeding.
  - Transparency bias (Kiparsky 1971) – this favors interactions that apply processes in an order that is transparent from the surface form (i.e. there are no surface violations of rules that applied somewhere in the grammar). Out of the interactions we just discussed, this bias favors feeding and bleeding.

	Application	**No Application**
Transparent	Feeding	Bleeding
Opaque	Counterbleeding	Counterfeeding

Recent approaches to interaction learning

- Experimental
  - Kim (2012) ran an artificial language learning experiment and found results that seem to suggest the opposite of a transparency bias (i.e. a bias *for* opaque interactions).
    - This study's bearing on learning biases is questionable since it primarily tested participants' choice of interaction rather than learnability.
    - In addition, this paper only qualitatively examined the experimental results, rather than running a statistical test.
  - Brooks et al. (2013) found a bias away from interacting in general.
    - Although this study also looked at participants' choice of pattern rather than overall learnability.
- Computational
  - Jarosz (2016) used an Expectation-Driven learning model in Harmonic Serialism with serial markedness constraints.
    - She found both transparency and application biases, depending on the kind of learning data that the model's trained on.
  - Nazarov and Pater (to appear) used a Maximum Entropy Stratal OT learner.
    - They found a transparency bias, but did not test for an application bias.
  - Rasin et al. (2017) used a minimum description length learner to learn a combination of ordered, SPE-style rules and the underlying representations for a toy language.
    - I don't have any results for this learner yet, but I plan on testing it soon.

2. Experimental data

Experiment design

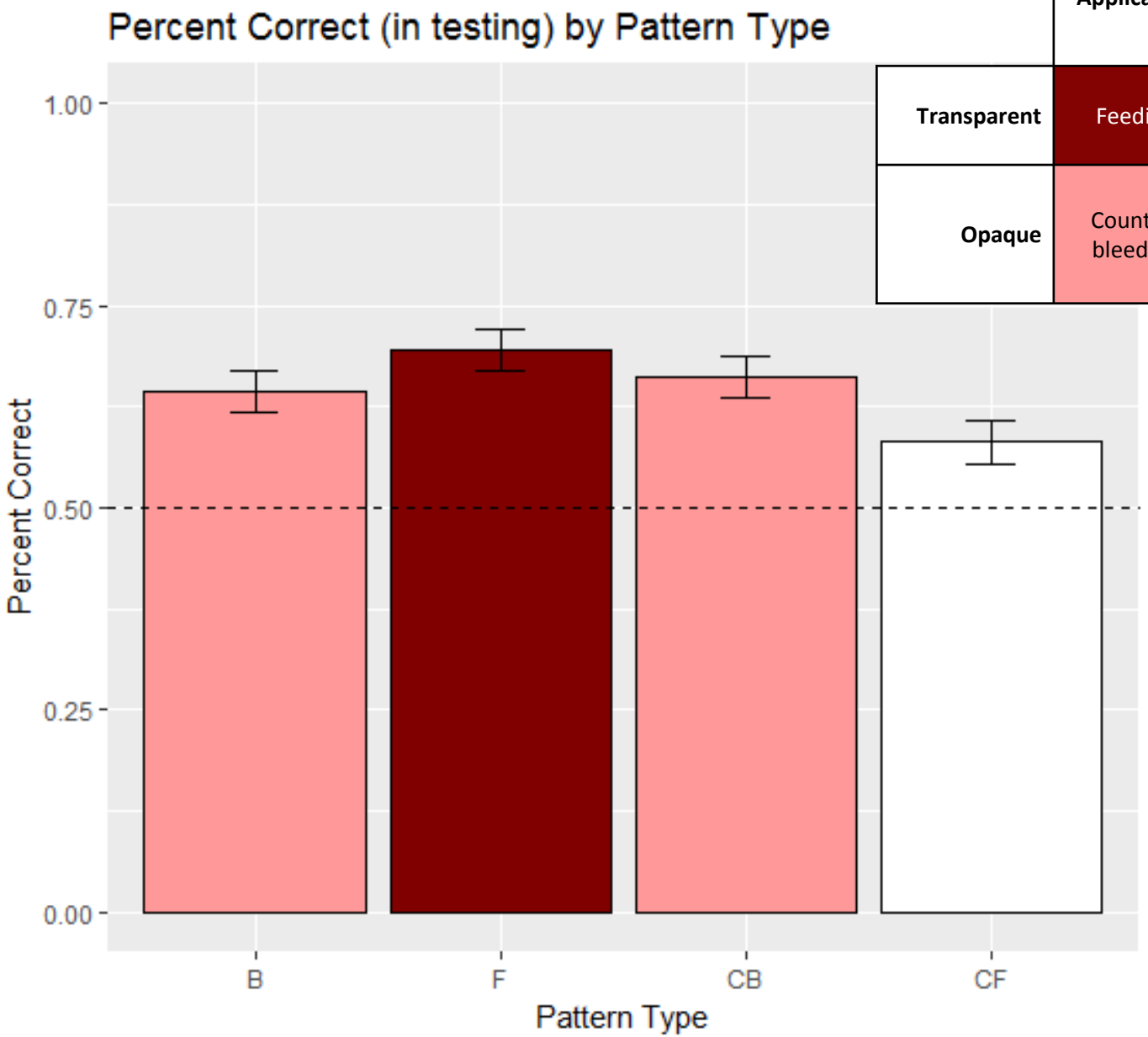
- To see whether the application and transparency biases affect language learning, I ran an artificial language experiment that directly tested the learnability different phonological interactions.
- Participants (N=48) were trained on 1 of the 4 languages that were previously discussed.
  - Coronals palatalized before [i] and vowels were deleted before other vowels.
- Stems in the languages all followed the regular expression /[iu][mnl][iau][tdkg]/. Two suffixes were added to stems, either individually or as a pair:

	/t/ + /-i/	/t/ + /-a/	/t/ + /-i/ and /-a/	/k/ + …
Bleeding	[tʃi]	[ta]	/tia/ → [ta]	[ki], [ka], [ka]
Feeding	[tʃi]	[ta]	/tai/ → [tʃi]	[ki], [ka], [ki]
Counterbleeding	[tʃi]	[ta]	/tia/ → [tʃa]	[ki], [ka], [ka]
Counterfeeding	[tʃi]	[ta]	/tai/ → [ti]	[ki], [ka], [ki]

- Training consisted of trials that asked participants to apply 1-2 affixes to a stem in response to a picture.
  - Affixation created plural and diminutive forms
  - Forced choice between adding morphemes in a phonologically licit and phonologically illicit way
  - After they made a choice, feedback was given in the form of “Correct!” or “Incorrect!”
- After training, they went through a test phase that was identical, except for a lack of feedback.

Experiment Results

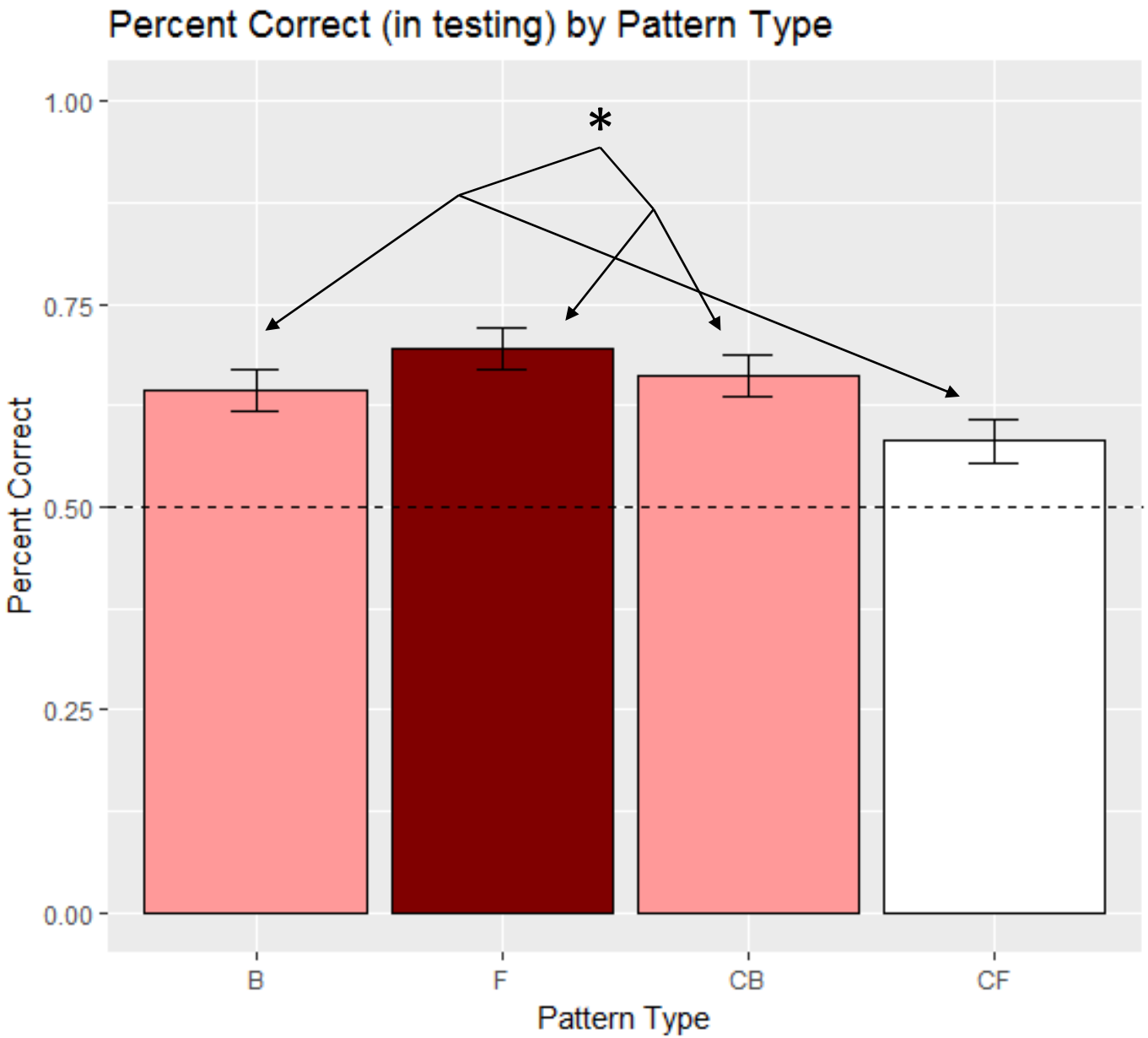
- The results on the right show that participants' accuracy was ordered in the same order that Kiparsky's (1968, 1971) biases would predict.
  - Error bars are 95% confidence intervals
  - This figure excludes data from a separate condition that included less explicit instructions for the participants.
- Feeding > Bleeding, C.B. > C.F.
- I also ran a mixed effects logistic regression model that supported this result.
  - Main effects were transparency, application, palatalization, and deletion (the latter two represent which kind of trial subjects' responses were for).
  - Random effects of subject and item on the intercepts were included.
- The main effects of application and transparency were both significant (p=.025 and .019, respectively).
- The interaction of application and transparency was not.



	Application	No Application
Transparent	Feeding	Bleeding
Opaque	Counter-bleeding	Counter-feeding

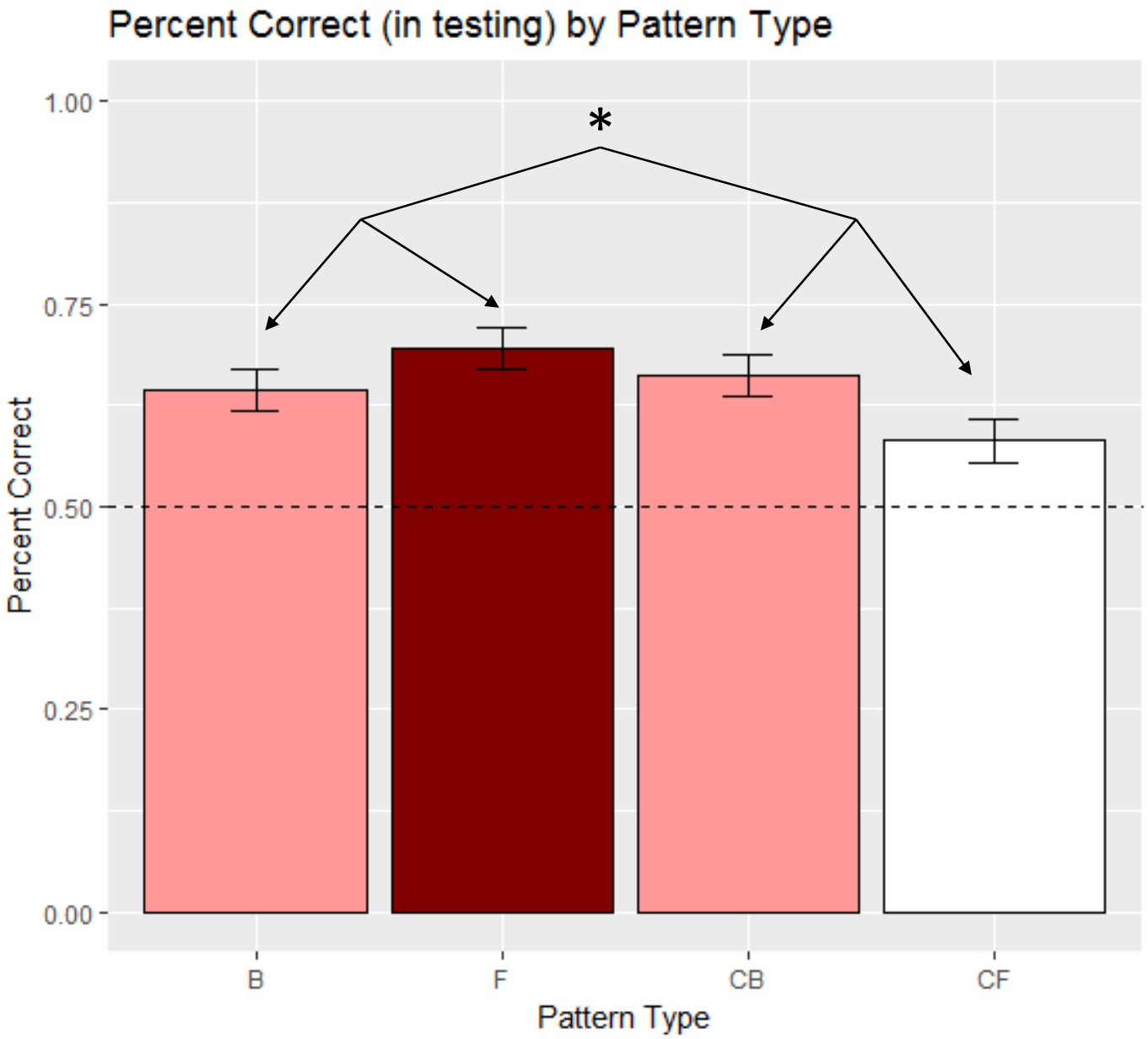
Experiment Results

- The results on the right show that participants' accuracy was ordered in the same order that Kiparsky's (1968, 1971) biases would predict.
  - Error bars are 95% confidence intervals
  - This figure excludes data from a separate condition that included less explicit instructions for the participants.
- Feeding > Bleeding, C.B. > C.F.
- I also ran a mixed effects logistic regression model that supported this result.
  - Main effects were transparency, application, palatalization, and deletion (the latter two represent which kind of trial subjects' responses were for).
  - Random effects of subject and item on the intercepts were included.
- The main effects of application and transparency were both significant (p=.025 and .019, respectively).
- The interaction of application and transparency was not.



Experiment Results

- The results on the right show that participants' accuracy was ordered in the same order that Kiparsky's (1968, 1971) biases would predict.
  - Error bars are 95% confidence intervals
  - This figure excludes data from a separate condition that included less explicit instructions for the participants.
- Feeding > Bleeding, C.B. > C.F.
- I also ran a mixed effects logistic regression model that supported this result.
  - Main effects were transparency, application, palatalization, and deletion (the latter two represent which kind of trial subjects' responses were for).
  - Random effects of subject and item on the intercepts were included.
- The main effects of application and transparency were both significant (p=.025 and .019, respectively).
- The interaction of application and transparency was not.



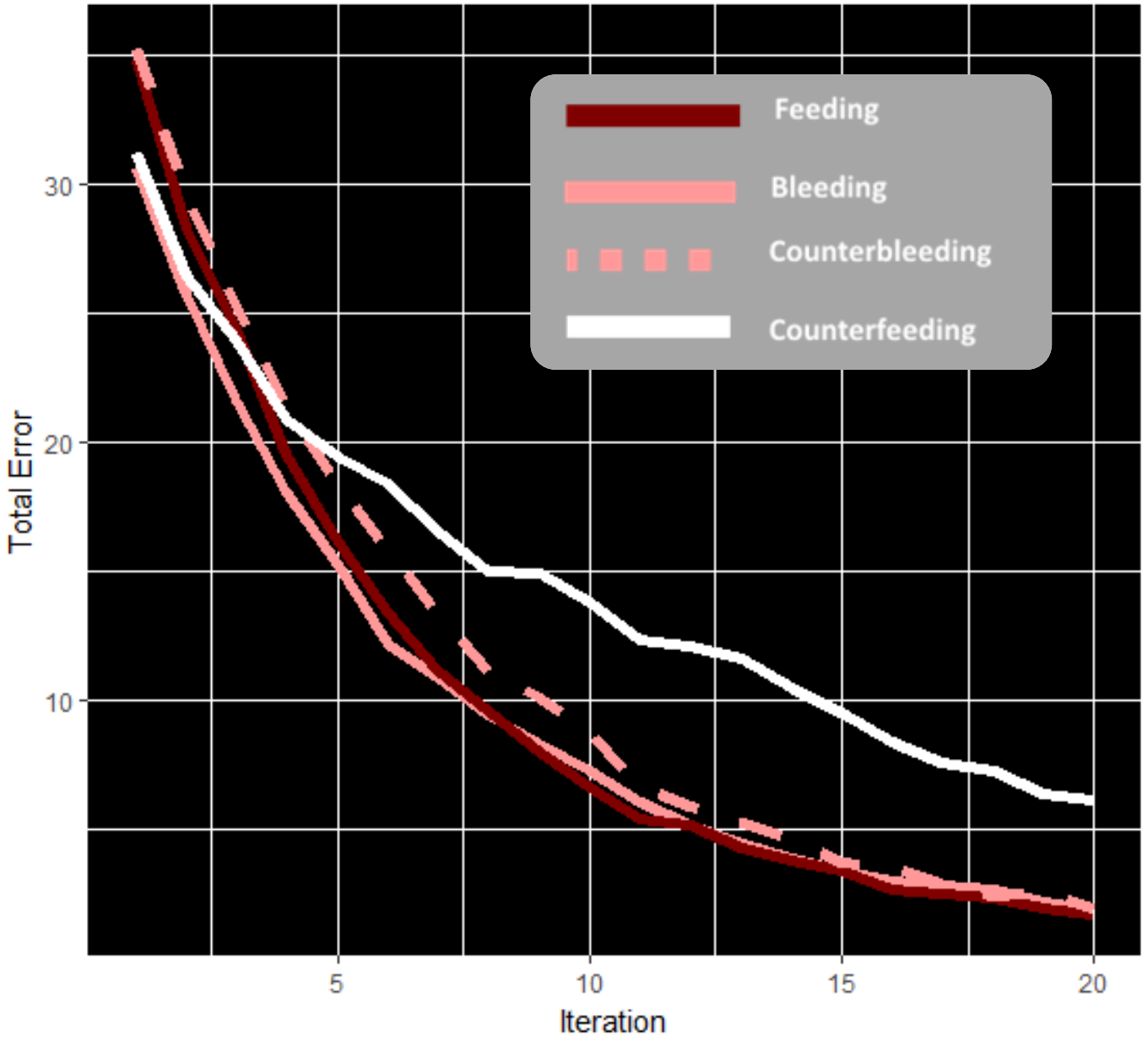
3. Comparing the models' performance

Expectation-Driven Learning in HS

- Jarosz (2016) used an expectation-driven harmonic serialism learner (EDL-HS; Jarosz 2015) to model the acquisition of different phonological interactions.
  - To represent opacity in HS, she used Serial Markedness Constraints (Jarosz 2013).
  - These allow the grammar to prefer derivations with a particular ordering for the application of markedness constraints.
  - The model learns by finding the probability distribution over pairwise constraint rankings that is most likely to have produced the training data.
- Jarosz (2016) found that skewing the data in different directions could cause the learner to have either of the Kiparskian biases.
  - When evidence of the interaction was high relative to evidence of individual rules (Jarosz's HI condition), there was an application bias.
  - When evidence of the individual rules was high relative to evidence of the interaction (Jarosz's LO condition), there was a transparency bias.
- When the learning data was balanced (Jarosz's UNI condition), counterfeeding was disfavored, but feeding had no advantage over bleeding and counterbleeding.

EDL-HS Results

- I ran the EDL-HS learner for 20 separate runs on each kind of interaction (frequency for each learning datum=18).
  - The figure on the right shows the average error for each iteration of the learning algorithm.
- Much like the subjects in my experiment, the learner struggled the most with the counterfeeding interaction.
- However, unlike humans, the learner failed to learn feeding better than bleeding and only had a slight advantage for feeding over counterbleeding.

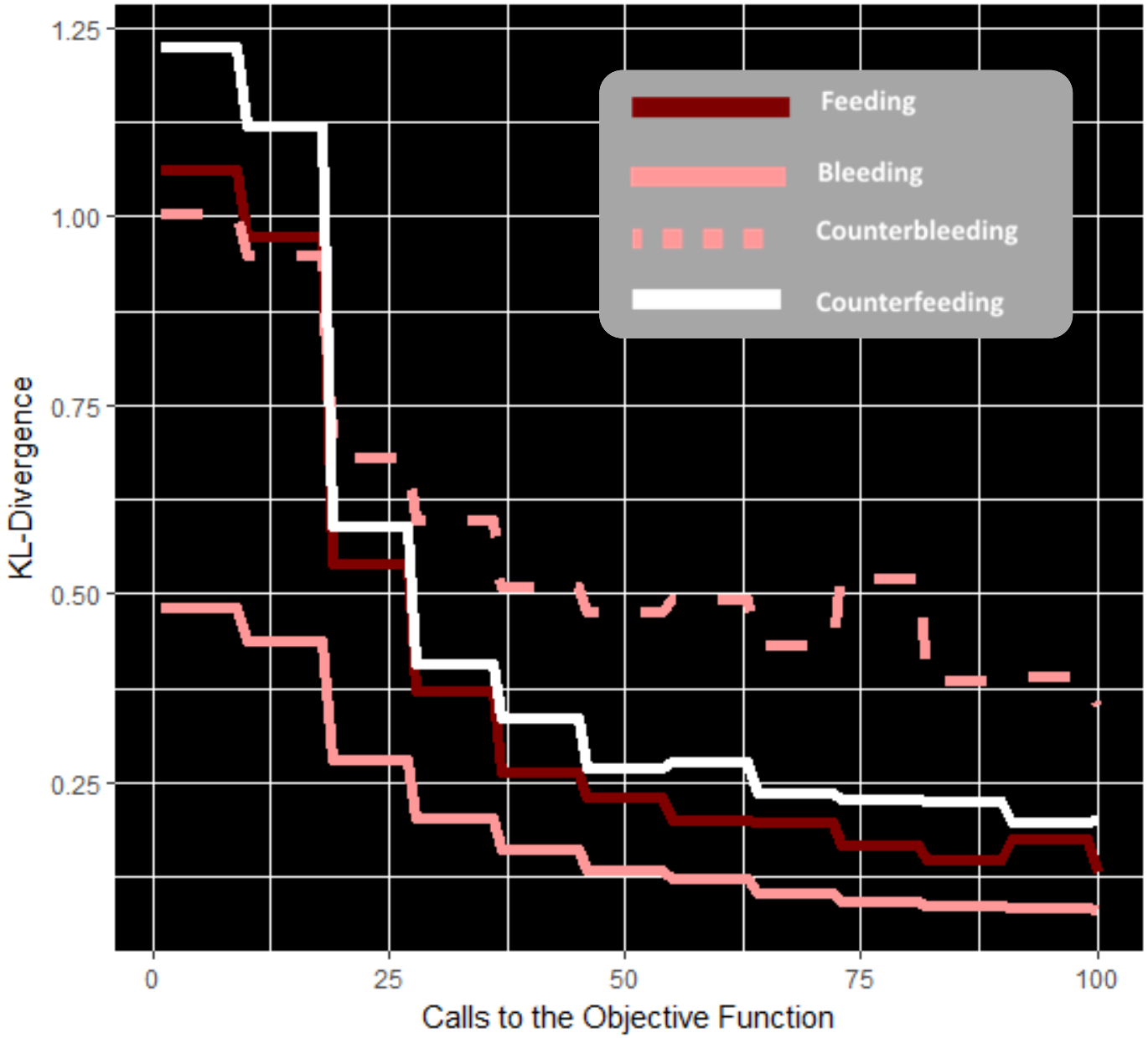


Stratal MaxEnt Learner

- Nazarov and Pater (to appear) used a Maximum Entropy Stratal OT learner to model the acquisition of bleeding and counterbleeding interactions.
  - Stratal OT (Booij 1996, Kiparsky 2000) is a framework that uses multiple, independent strata of constraints to represent a phonological derivation. It can represent all four kinds of interactions discussed here because of the independence of rankings across strata (although see McCarthy 2007 on the limitations of this theory).
  - The learner from Nazarov and Pater (to appear) is a MaxEnt (Goldwater and Johnson 2003, Hayes and Wilson 2008) version of this framework (see Staubs and Pater 2016 for more on learning MaxEnt grammars in a derivational framework).
  - The model learns by minimizing the KL-divergence between the probabilities of each UR→SR mapping in the training data and the probabilities given to these mappings by the model's grammar.
- Nazarov and Pater (to appear) found that bleeding was learned more reliably by their learner than counterbleeding.
  - Learnability was measured by the proportion of runs in which the learner converged on the correct grammar.
  - When the learner doesn't converge, it's because it becomes caught in a local minimum that the optimization algorithm (L-BFGS) can't get out of.

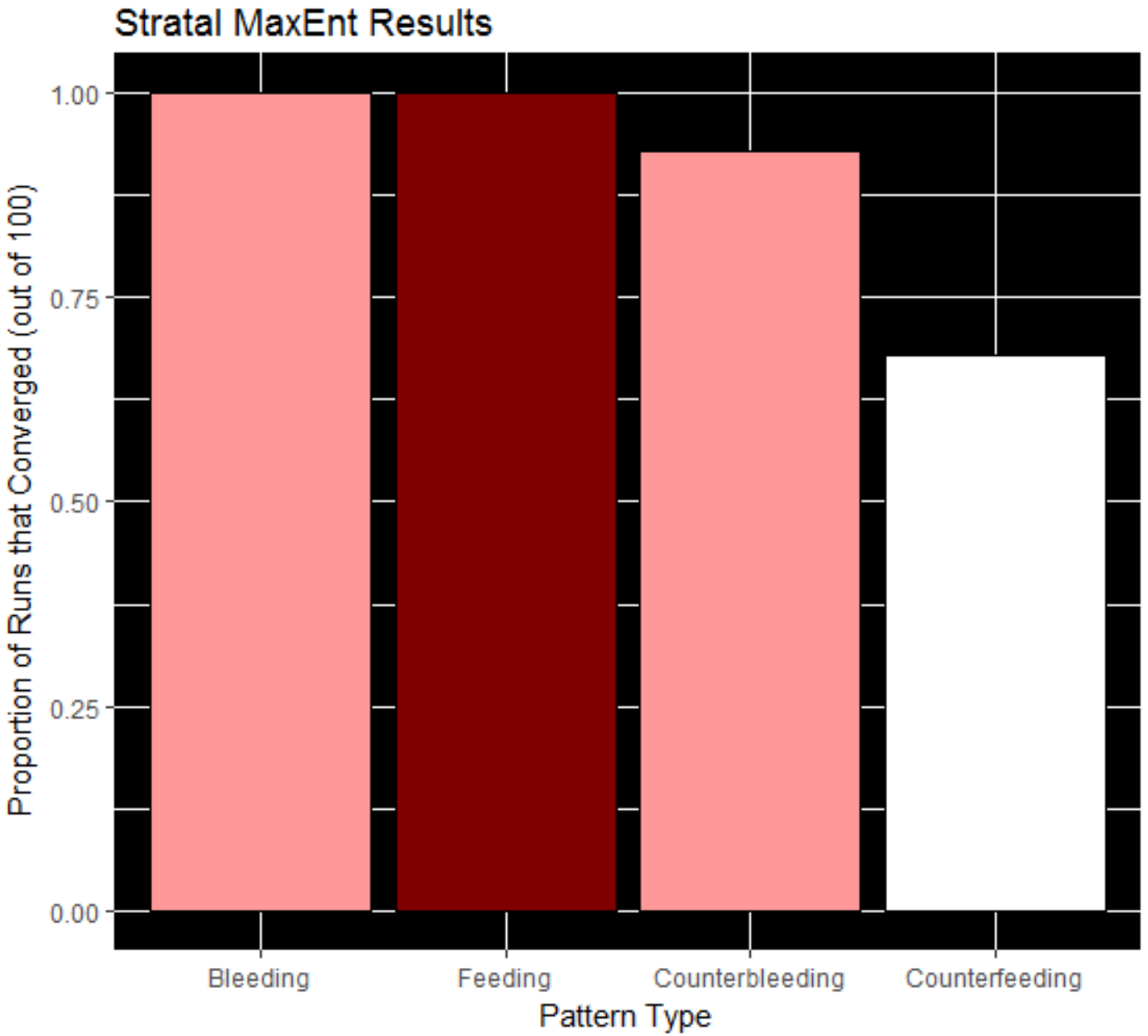
Stratal MaxEnt Results

- I ran the Stratal MaxEnt learner (σ^2 = 4000, number of strata=2) for 20 runs on each kind of interaction.
  - The figure on the right shows the average KL-Divergence on each call to the objective function (i.e. the function that was being minimized by the optimizer).
- The learning curves on the right show a preference for **B>F,CF>CB**.
  - Which does not correlate well with the human behavior.
  - Learning curves aren't the best here since



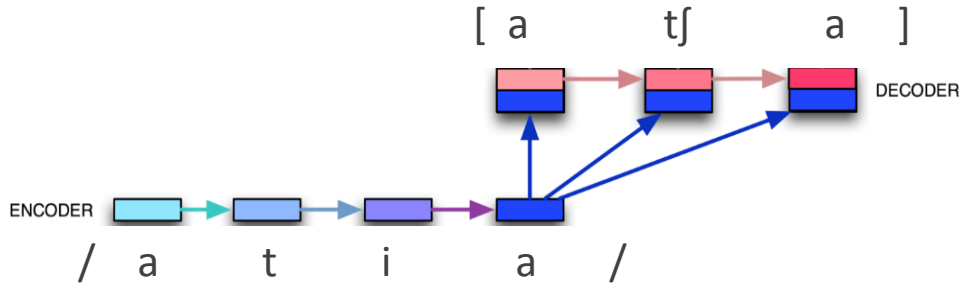
Stratal MaxEnt Results

- I ran the Stratal MaxEnt learner (σ^2 = 4000, number of strata=2) for 20 runs on each kind of interaction.
  - The figure on the right shows the average KL-Divergence on each call to the objective function (i.e. the function that was being minimized by the optimizer).
- The learning curves on the right show a preference for **B>F,CF>CB**.
  - Which does not correlate well with the human behavior.
  - Learning curves aren't the best here since
- If instead you use proportion of runs that converged you get **F,B>CB>CF**.
  - This is the metric that Nazarov and Pater (to appear) use in their paper.



Sequence-to-Sequence Neural Net

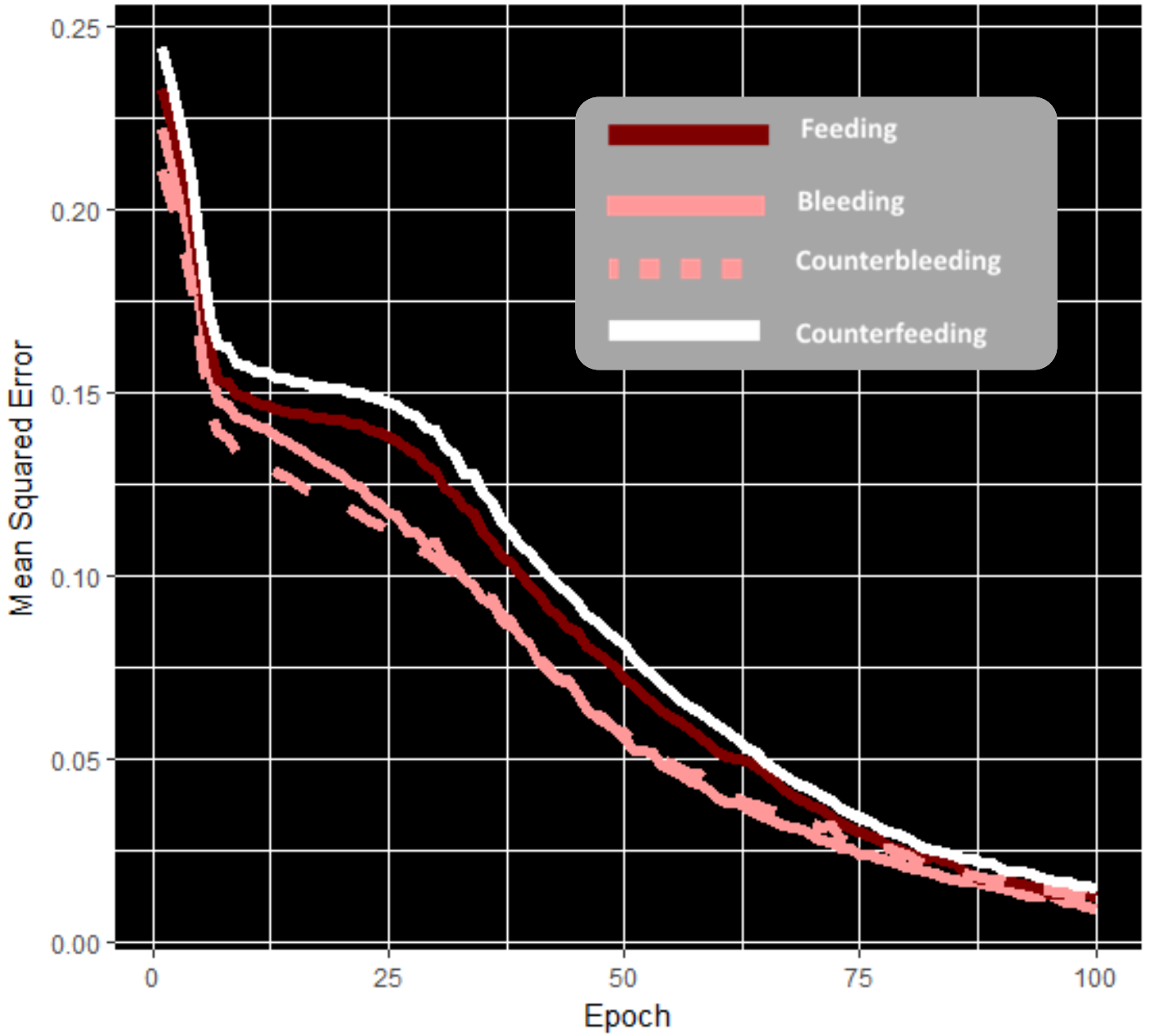
- Kirov (2017) used a Sequence-to-Sequence (Seq2Seq) neural network to model alternations.
  - Seq2Seq neural networks are made up of two parts: and encoder and a decoder.
  - Each part is a separate recurrent neural network, with the encoder's output acting as the decoder's input. (figure adapted from https://medium.com/@devnag/seq2seq-the-clown-car-of-deep-learning-f88e1204dac3)



- He found that the model correlated well with human phonological learning.
- While Kirov (2017) didn't use his model to test opacity, it's highly flexible and can easily be used for such a task.
  - One difference in the simulations I'll be running is that while Kirov's model mapped from singular to plural, mine will map from UR to SR.
  - This is a result of multiple affixes being used in my experiment, something Kirov never deals with in his simulations.

Seq2Seq Results

- I ran the Seq2Seq learner (number of hidden layers=4, number of nodes per hidden layer=100, number of epochs=100) for 20 runs on each kind of interaction.
  - The figure to the right shows the average mean squared error after 15 epochs.
- Counterfeeding was disfavored, like the subjects in my experiment.
- However, the model had trouble learning the feeding pattern well.
  - This is in direct contrast to the experimental human results, which found feeding to be the most learnable kind of pattern.



4. Conclusions

Which learner predicted the data best?

- Out of these three learners, the Expectation-driven Learning with SM constraints had the best learning curves.
  - Although learning curves might not have been the best way to measure every learner's performance.
- The results suggest that most models can disfavor counterfeeding like my subjects did.
  - i.e. they displayed an *application* and *transparency* bias **against** counterfeeding.
- Although they all struggled with favoring the feeding interactions as much as humans seem to do.
  - i.e. they **didn't** display an *application* and *transparency* bias **for** the feeding interaction.
  - This was especially true of the neural network, which found feeding to be *more* difficult than bleeding/counterbleeding.

Future work

- UR-learning could be driving some of the biases observed in the human data.
  - None of the simulations ran here involved UR learning.
  - A UR-learning module for the EDL-HS learner is in the process of being made (Jarosz, personal communication).
  - The neural net could be engineered to do UR-learning in a number of ways (e.g. by mapping from semantics to surface forms like the subjects were tasked with doing; see Gasser 1993 for a similar approach).
  - The Stratal MaxEnt learner could be adapted to learn UR's as well (see Eisenstat 2009, Pater et al. 2012, and O'Hara 2017 for various methods of learning UR's in a MaxEnt framework).
- Rasin et al.'s (2017) MDL-based opacity learner can also tested to see if it models these biases.
  - Since this model already learns UR's, it would also be a first step toward modeling opacity biases.
- My experiment and simulations only test opacity on the environment, but there's reason to believe opacity on the focus could have different biases.
- What kind of data are infants getting when they learn these interactions?
  - Does this experiment and do these simulations accurately reflect that?
  - Or would Jarosz's (2016) skewed datasets (with more/less evidence for the interaction) be more accurate?

References

Baković, Eric (2011). Opacity and ordering. In John Goldsmith, Jason Riggle, and Alan Yu (eds.), *The Handbook of Phonological Theory (2nd ed)*. Malden, MA: Wiley-Blackwell. 40-67.

Brooks, K. M., Pajak, B., & Bakovic, E. (2013). *Learning biases for phonological interactions*. Poster presented at *2013 Meeting on Phonology.*

Rasin, Ezer, Iddo Berger, Nur Lan, and Roni Katzir (2017). *Acquiring opaque phonological interactions using minimum description length*. Poster preseted at the *2017 Annual Meeting on Phonology*

Goldwater, Sharon, and Mark Johnson (2003). Learning OT Constraint Rankings Using a Maximum Entropy Model. In Jennifer Spenader, Anders Eriksson, and Osten Dahl (eds.), *Proceedings of the Stockholm Workshop on Variation within Optimality Theory.* 111–20.

Hayes, Bruce, and Colin Wilson (2008). A Maximum Entropy Model of Phonotactics and Phonotactic Learning. *Linguistic Inquiry 39,* 379–440.

Jarosz, Gaja (2015). *Expectation Driven Learning of Phonology*. Ms., University of Massachusetts Amherst.

Jarosz, Gaja (2014). Serial Markedness Reduction. *Proceedings of the 2013 Meeting on Phonology 1(1)*, Amherst, MA.

Jarosz, Gaja (2016). Learning Opaque and Transparent Interactions in Harmonic Serialism. In *Proceedings of the Annual Meetings on Phonology (Vol. 3)*.

Kim, Yun Jung (2012). Do learners prefer transparent rule ordering? An artificial language learning study. In *Proceedings from the Annual Meeting of the Chicago Linguistic Society, 48(1)*, 375-386. Chicago Linguistic Society.

Kiparsky, Paul (1968). Linguistic universals and linguistic change. In Emmon Bach & Robert T. Harms (eds.), *Universals in linguistic theory*. New York : Holt, Reinhart & Winston. 170–202.

Kiparsky, Paul (1971). Historical linguistics. In W. O. Dingwall (ed.) *A Survey of Linguistic Science*. College Park: University of Maryland Linguistics Program. 576-642.

Kirov, Christo (2017). *Recurrent Neural Networks as a Strong Domain-General Baseline for Morpho-Phonological Learning.* Poster presented at the *2017 Meeting of the Linguistic Society of America*.

McCarthy, John J. 2007. *Hidden generalizations: phonological opacity in Optimality Theory*. London: Equinox Press.

Nazarov, Aleksei and Joe Pater (to appear). Learning opacity in Stratal Maximum Entropy Grammar. *Phonology*.

Pater, Joe, Karen Jesney, Robert Staubs, and Brian Smith (2012). Learning Probabilities over Underlying Representations. In *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology*, 62–71. Association for Computational Linguistics.

Rahman, Fariz (2016). *Seq2Seq*. Documentation and download available at: https://github.com/farizrahman4u/seq2seq

Staubs, Robert, and Pater, Joe (2016). Learning serial constraint-based grammars. In John McCarthy and Joe Pater (eds.), *Harmonic Grammar and Harmonic Serialism*. London: Equinox Press.

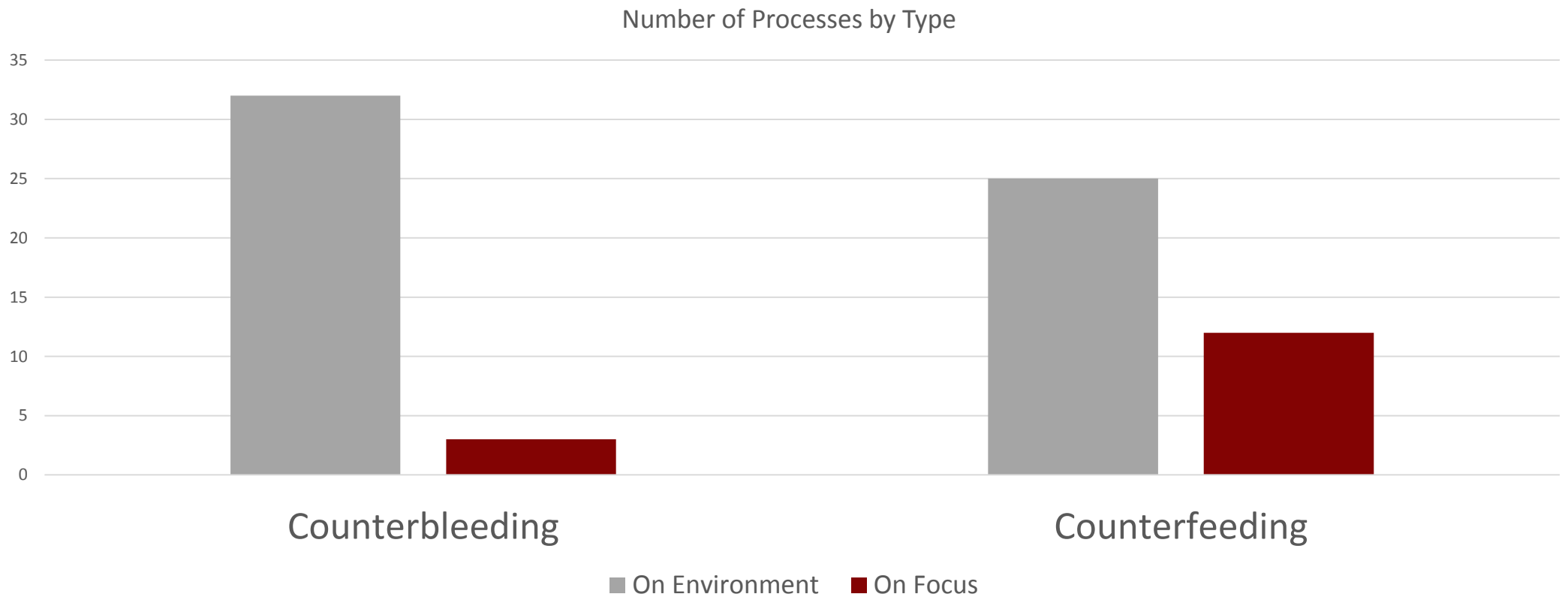
Acknowledgments

Thanks to the UMass Sound Workshop and Grant Group for helpful feedback on this presentation. Alexei Nazarov and Christo Kirov both provided helpful email correspondence about their respective learning models. And a big thanks to Joe Pater and Gaja Jarosz for advising me throughout this project. Experiment subjects were paid using funds from NSF Grant #BCS-1650957.

Appendix

I. Typological Data

Typological data for opaque interactions



(Data collection funded by NSF grant BCS-424077; for more information, contact Joe Pater)

II. Experiment Details

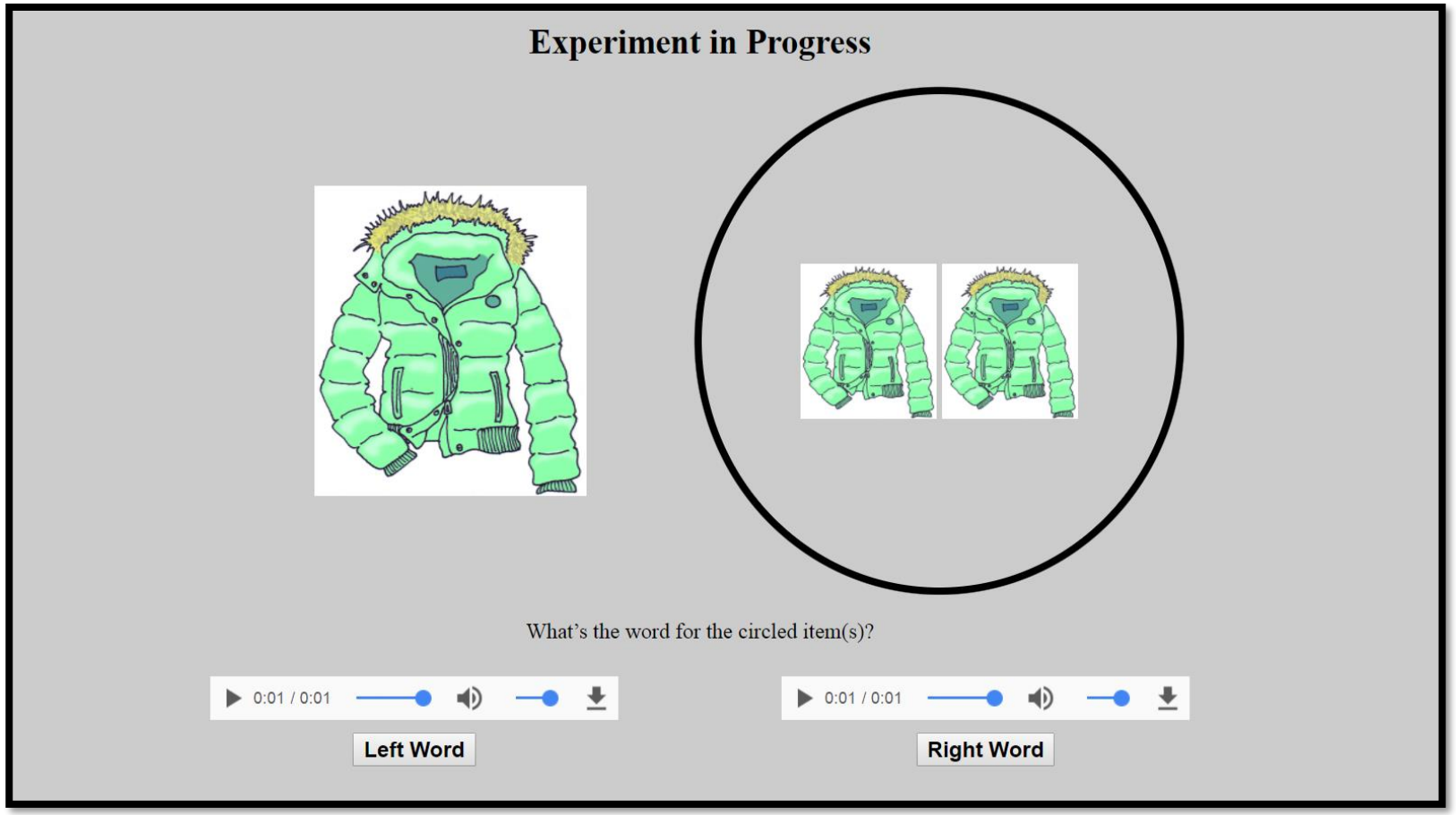
Methodological details

- **Stimuli:**
  - There were 72 total stems, divided between training and testing.
  - Subjects saw every stem with every possible combination of affixes (plural, diminutive, plural+diminutive), for a total of 216 trials (108 per phase).
  - No stems were repeated across testing and training.
  - The affixes were [i] and [a]. Their definitions (i.e. diminutive and plural) were counterbalanced.
  - Stems were recorded in a sound attenuated booth by the author (a native speaker of Southern American English).
  - Visual stimuli accompanied each trial to demonstrate semantic information (plural, diminutive,or plural+diminutive). Each picture was randomly paired with a stem.
- **Participants:**
  - 48 participants were recruited on Mechanical Turk.
  - They were limited to people whose IP address was in the US and who had more than 1000 accepted HITS, with a percentage of acceptance >= 85.
  - Subjects were all self-reported native English speakers.

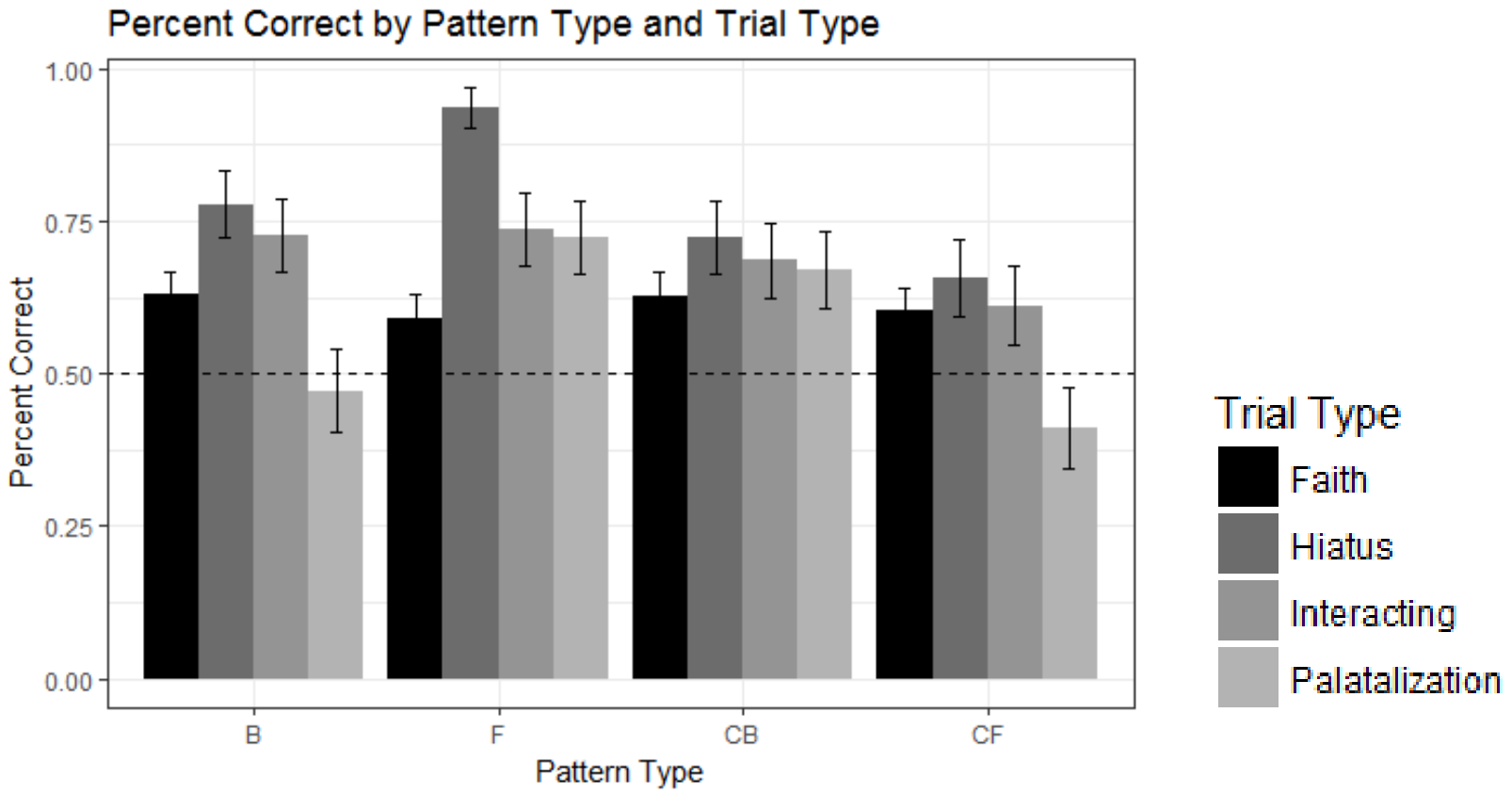
Methodological details

- **Procedure**
  - Subjects were first presented with instructions stating that they were going to learn an alien language.
  - They were told that “languages often have rules that apply when a suffix is added to a word. For example, the 'f' in 'hoof' changes to a 'v' in the plural form ('hooves’),” and were encouraged to find similar rules in the alien language.
  - They were tested on three practice questions in English and had to answer these correctly before they were allowed to proceed.
    - Question 1: they were shown a picture of a dog and a cat, with the cat circled and had to choose an audio file that corresponded to the circled picture (choice of ‘cat’ vs. ‘dog’).
    - Question 2: they were shown a picture of two locks and a picture of one lock. The picture of two locks was circled, and they again had to choose an audio file that corresponded to the circled picture (choice of ‘lock’ vs. ‘locks’).
    - Question 3: they were shown a picture of a statue and a smaller statue. The smaller statue was circled. They had to choose an audio file that corresponded to the circled picture (choice of ‘statue’ vs. ‘statuette’).
  - They then began the training phase. Here they again had to choose audio files that corresponded to the circled picture that they were shown.
    - This circled presentation format was necessary to demonstrate the relative size of the diminutive pictures.
  - After training and testing, they answer a series of demographic questions and questions about their experience in the experiment.

Screenshot of experiment software



Experimental results by trial type



Logistic regression results

- I ran a mixed effects regression analysis, using the glmer() function in R.
  - My main effects were Transparent, Application, Deletion, and Palatalization. I also had random effects of subjects and stimuli on the intercepts.
  - glmer(Correct ~ Transparent*Application*Deletion*Palatalization+(1|Subject) +(1|Stimulus), family = "binomial", glmerControl(optimizer="bobyqa", optCtrl = list(maxfun = 100000)), data = exp_data)

	+Delet.	-Delet.
+Palat.	Interacting	Palatal.
-Palat.	Hiatus	Faith

	+Transpar.	-Transpar.
+Applic.	Feeding	C.B.
-Applic.	Bleeding	C.F.

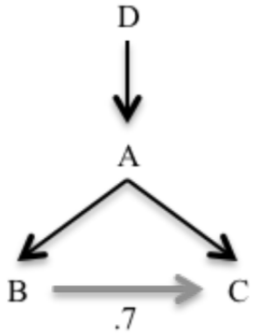
	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	0.84542	0.11646	7.259	3.89e-13	***
Transparent	**0.26731**	**0.11429**	**2.339**	**0.019342**	*
Application	**0.25538**	**0.11426**	**2.235**	**0.025415**	*
Deletion	0.39945	0.04411	9.057	< 2e-16	***
Palatalization	-0.20457	0.04397	-4.652	3.29e-06	***
Transparent:Application	0.02355	0.11513	0.205	0.837909	
Transparent:Deletion	0.18515	0.03845	4.816	1.47e-06	***
Application:Deletion	-0.01127	0.03845	-0.293	0.769538	
Transparent:Palatalization	-0.06703	0.03835	-1.748	0.080468	.
Application:Palatalization	0.06670	0.03837	1.738	0.082153	.
Deletion:Palatalization	-0.10457	0.04397	-2.378	0.017388	*
Transparent:Application:Deletion	0.06628	0.04095	1.619	0.105515	
Transparent:Application:Palatalization	-0.06774	0.04086	-1.658	0.097386	.
Transparent:Deletion:Palatalization	-0.13848	0.03837	-3.609	0.000308	***
Application:Deletion:Palatalization	-0.24473	0.03841	-6.372	1.86e-10	***
Transparent:Application:Deletion:Palatalization	-0.11768	0.04090	-2.877	0.004011	**

III. Expectation Driven Learner

Expectation-Driven Learning

- Jarosz (2015) introduced an expectation-driven learner for acquiring hidden structure in phonological patterns.
- This learner uses probabilistic constraint rankings to represent the grammar (figure from Jarosz 2015):

	A	B	C	D
A		1	1	0
B	0		.7	0
C	0	.3		0
D	1	1	1	

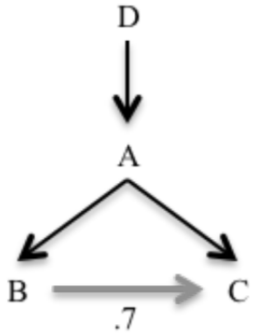


Expectation-Driven Learning

- Jarosz (2015) introduced an expectation-driven learner for acquiring hidden structure in phonological patterns.
- This learner uses probabilistic constraint rankings to represent the grammar (figure from Jarosz 2015):

	A	B	C	D
A		1	1	0
B	0		.7	0
C	0	.3		0
D	1	1	1	

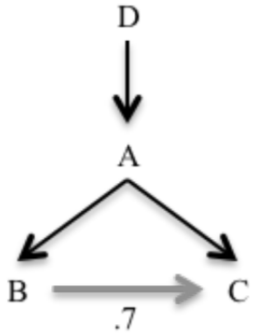
Constraints



Expectation-Driven Learning

- Jarosz (2015) introduced an expectation-driven learner for acquiring hidden structure in phonological patterns.
- This learner uses probabilistic constraint rankings to represent the grammar (figure from Jarosz 2015):

	A	B	C	D
A		1	1	0
B	0		.7	0
C	0	.3		0
D	1	1	1	

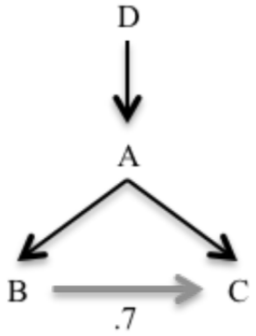


Probability that Constraint 1 >> Constraint 2

Expectation-Driven Learning

- Jarosz (2015) introduced an expectation-driven learner for acquiring hidden structure in phonological patterns.
- This learner uses probabilistic constraint rankings to represent the grammar (figure from Jarosz 2015):

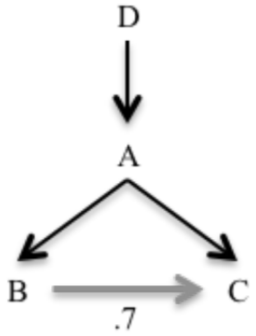
	A	B	C	D
A		1	1	0
B	0		.7	0
C	0	.3		0
D	1	1	1	



Expectation-Driven Learning

- Jarosz (2015) introduced an expectation-driven learner for acquiring hidden structure in phonological patterns.
- This learner uses probabilistic constraint rankings to represent the grammar (figure from Jarosz 2015):

	A	B	C	D
A		1	1	0
B	0		.7	0
C	0	.3		0
D	1	1	1	

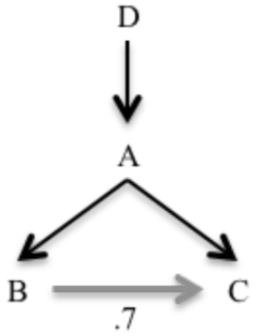


- To learn the ranking, for example, between A and B:
  - $P(A>>B \mid data, grammar) = \frac{P(data \mid A>>B, grammar) * P(A>>B \mid grammar)}{(data \mid grammar)}$
  - Where "data" is the learning data and "grammar" is the current ranking of all the constraints.

Expectation-Driven Learning

- Jarosz (2015) introduced an expectation-driven learner for acquiring hidden structure in phonological patterns.
- This learner uses probabilistic constraint rankings to represent the grammar (figure from Jarosz 2015):

	A	B	C	D
A		1	1	0
B	0		.7	0
C	0	.3		0
D	1	1	1	



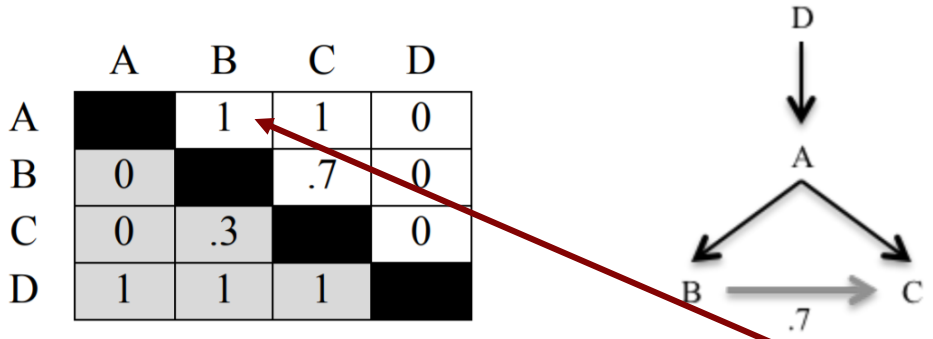
- To learn the ranking, for example, between A and B:
  - P(A>>B | data, grammar) = $\frac{P(\text{data} \mid A>>B, \text{grammar}) * P(A>>B \mid \text{grammar})}{(\text{data} \mid \text{grammar})}$

Measured by seeing how often this ranking produces each output in the learning data

  - Where "data" is the learning data and "grammar" is the current ranking of all the constraints.

Expectation-Driven Learning

- Jarosz (2015) introduced an expectation-driven learner for acquiring hidden structure in phonological patterns.
- This learner uses probabilistic constraint rankings to represent the grammar (figure from Jarosz 2015):



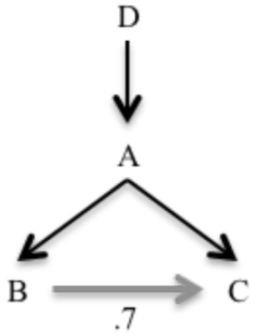
This is taken from the current grammar

- To learn the ranking, for example, between A and B:
  - P(A>>B | data, grammar) = $\frac{P(data \mid A>>B, grammar) * P(A>>B \mid grammar)}{(data \mid grammar)}$
  - Where "data" is the learning data and "grammar" is the current ranking of all the constraints.

Expectation-Driven Learning

- Jarosz (2015) introduced an expectation-driven learner for acquiring hidden structure in phonological patterns.
- This learner uses probabilistic constraint rankings to represent the grammar (figure from Jarosz 2015):

	A	B	C	D
A		1	1	0
B	0		.7	0
C	0	.3		0
D	1	1	1	



- To learn the ranking, for example, between A and B:
  - P(A>>B | data, grammar) = $\frac{\text{P(data | A>>B, grammar) * P(A>>B | grammar)}}{\text{(data | grammar)}}$

This is the sum of the numerator and its complement.

  - Where "data" is the learning data and "grammar" is the current ranking of all the constraints.

IV. SM Constraints

Serial Markedness Constraints

- Representing counterbleeding and counterfeeding in Harmonic Serialism requires an extra piece of machinery.
- Jarosz (2013) proposed Serial Markedness (SM) constraints for this purpose.
- These constraints specify which order markedness constraints are satisfied over the course of a derivation.
- Example (ranking is for counterbleeding; only faithful, counterbleeding, and bleeding candidates shown):

/tia/	SM(*VV, *ti)	SM(*ti, *VV)	*ti	*VV	MAX	IDENT
tia, [tia]	--, --	--, --	*, *	*, *	--, --	--, --
→tʃia, [tʃa]	--, --	-- , *	--, --	*, --	--, *	*, --
ta, [ta]	-- , *	--, --	--, --	--, --	*, --	--, --

Serial Markedness Constraints

- Representing counterbleeding and counterfeeding in Harmonic Serialism requires an extra piece of machinery.
- Jarosz (2013) proposed Serial Markedness (SM) constraints for this purpose.
- These constraints specify which order markedness constraints are satisfied over the course of a derivation.
- Example (ranking is for counterbleeding; only faithful, counterbleeding, and bleeding candidates shown):

Candidates in the first step of the derivation

/tia/	SM(*VV, *ti)	SM(*ti, *VV)	*ti	*VV	MAX	IDENT
tia, [tia]	--, --	--, --	*, *	*, *	--, --	--, --
➔ tʃia, [tʃa]	--, --	-- , *	--, --	*, --	--, *	*, --
ta, [ta]	-- , *	--, --	--, --	--, --	*, --	--, --

Serial Markedness Constraints

- Representing counterbleeding and counterfeeding in Harmonic Serialism requires an extra piece of machinery.
- Jarosz (2013) proposed Serial Markedness (SM) constraints for this purpose.
- These constraints specify which order markedness constraints are satisfied over the course of a derivation.
- Example (ranking is for counterbleeding; only faithful, counterbleeding, and bleeding candidates shown):

Candidates in the final step of the derivation

/tia/	SM(*VV, *ti)	SM(*ti, *VV)	*ti	*VV	MAX	IDENT
tia, [tia]	--, --	--, --	*, *	*, *	--, --	--, --
→ tʃia, [tʃa]	--, --	-- , *	--, --	*, --	--, *	*, --
ta, [ta]	--, *	--, --	--, --	--, --	*, --	--, --

V. MaxEnt Stratal Learner

MaxEnt stratal learner

- The learner maximizes the likelihood of the learning data by changing constraint weights (weights are analogous to OT constraint rankings).
  - Input tableaux give constraints violations for every input-output mapping.
  - Learning data is a table that gives probabilities for the different possible input-output mappings.
  - A regularization terms pressures constraints to have low weights.
- The learning algorithm minimizes KL-divergence between the probabilities predicted by the constraint weights and the input-output probabilities given to the learner.
  - The probability of a single input-output path is the product of each step in the derivation.
  - The probability of an input-output mapping is the sum of all the path probabilities that would end in that output.
- Both I and Nazarov and Pater (to appear) use “L-BFGS-B” optimization to find the ideal weight values.

MaxEnt Stratal Learning Example

UR	Stratum 1 Output	Stratum 2 Output (SR)
A	A	A
	B	B

- P(A→B) = ?

MaxEnt Stratal Learning Example

UR	Stratum 1 Output	Stratum 2 Output (SR)
A →	A	A
	B	B

- $P(A \rightarrow B) = P(A \rightarrow A)_{\text{Stratum 1}} \cdots$

MaxEnt Stratal Learning Example

UR	Stratum 1 Output	Stratum 2 Output (SR)
A	A	A
	B	B

- $P(A \rightarrow B) = P(A \rightarrow A)_{\text{Stratum 1}} * P(A \rightarrow B)_{\text{Stratum 2}} \cdots$

P(Path 1)

MaxEnt Stratal Learning Example

UR	Stratum 1 Output	Stratum 2 Output (SR)
A	A	A
	B	B

- $P(A \rightarrow B) = P(A \rightarrow A)_{\text{Stratum 1}} * P(A \rightarrow B)_{\text{Stratum 2}} + P(A \rightarrow B)_{\text{Stratum 1}} \ldots$

P(Path 1)

MaxEnt Stratal Learning Example

UR	Stratum 1 Output	Stratum 2 Output (SR)
A	A	A
	B	B

- $P(A \rightarrow B) = P(A \rightarrow A)_{\text{Stratum 1}} * P(A \rightarrow B)_{\text{Stratum 2}} + P(A \rightarrow B)_{\text{Stratum 1}} * P(B \rightarrow B)_{\text{Stratum 2}}$

P(Path 1) P(Path 2)

MaxEnt Stratal Learning Example

UR	Stratum 1 Output	Stratum 2 Output (SR)
A	A	A
	B	B

- $P(A \rightarrow B) = P(A \rightarrow A)_{\text{Stratum 1}} * P(A \rightarrow B)_{\text{Stratum 2}} + P(A \rightarrow B)_{\text{Stratum 1}} * P(B \rightarrow B)_{\text{Stratum 2}}$

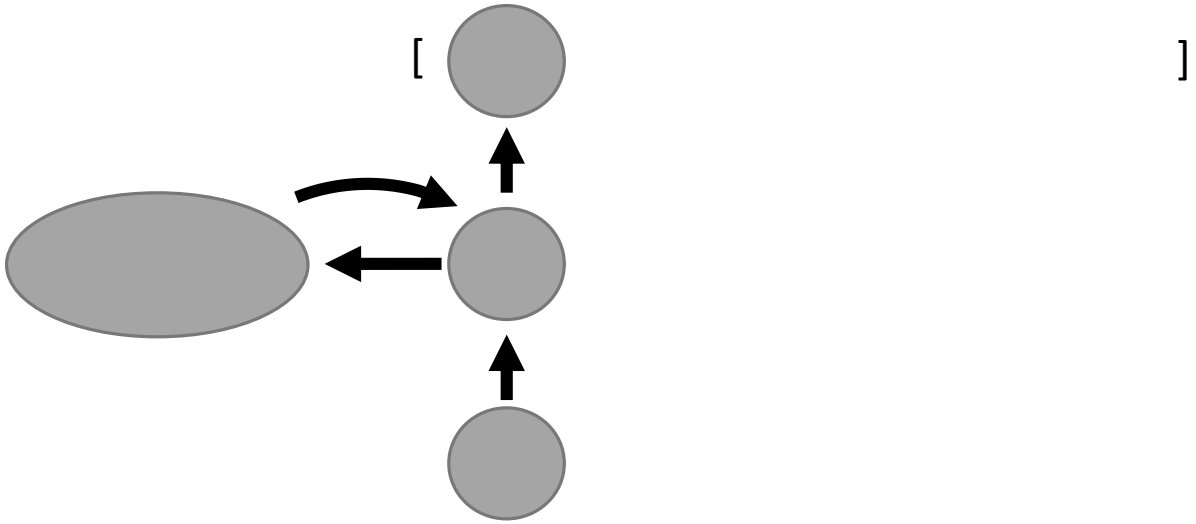
P(Path 1) P(Path 2)

- $P(A \rightarrow B) = P(\text{Path 1}) + P(\text{Path 2})$

VI. Recurrent Neural Networks

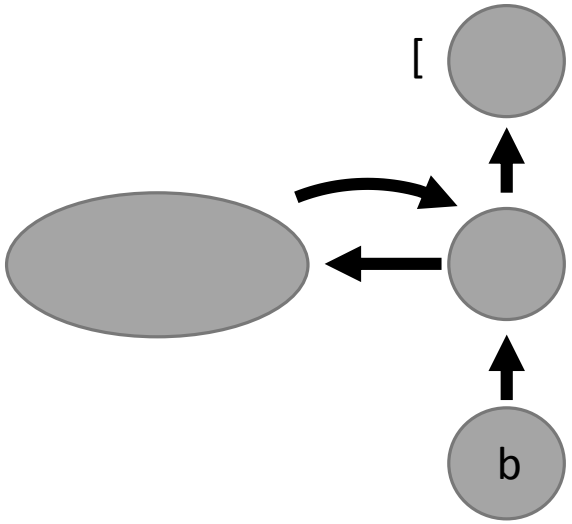
Recurrent Neural Networks

- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, a segment) at a time.



Recurrent Neural Networks

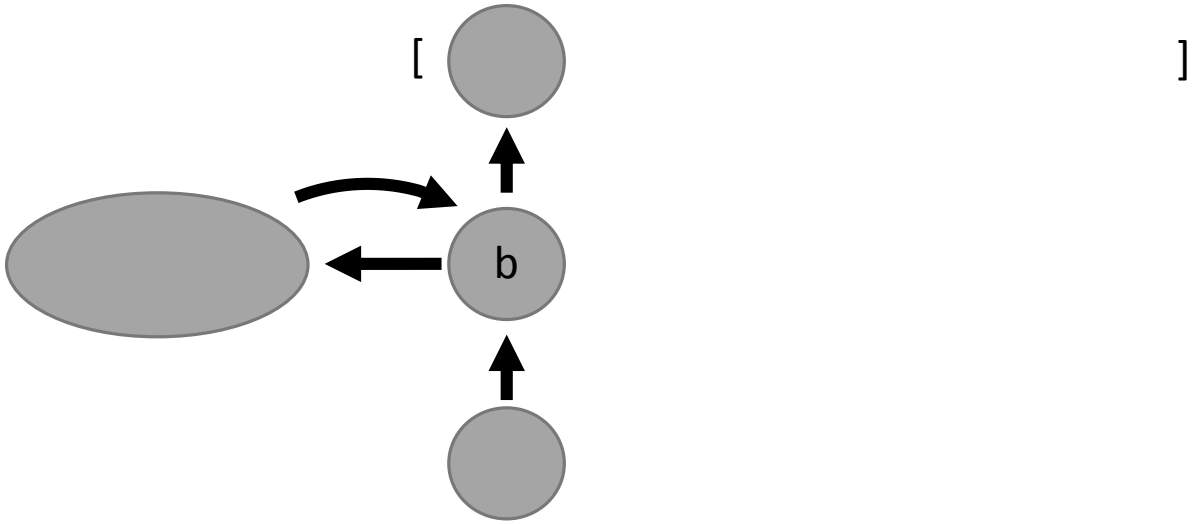
- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, segments) at a time.



/ b a t # /

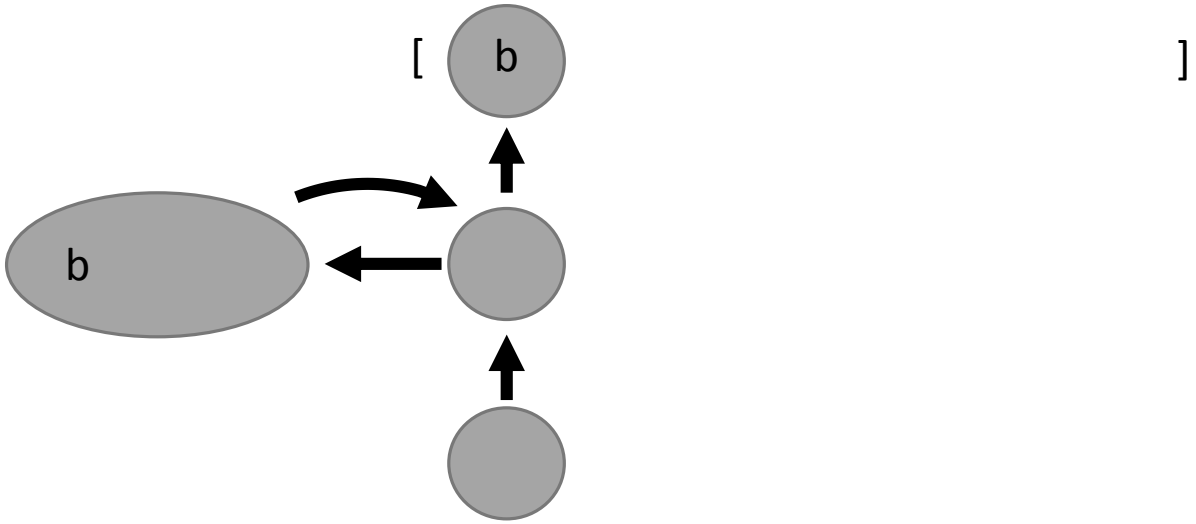
Recurrent Neural Networks

- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, segments) at a time.



Recurrent Neural Networks

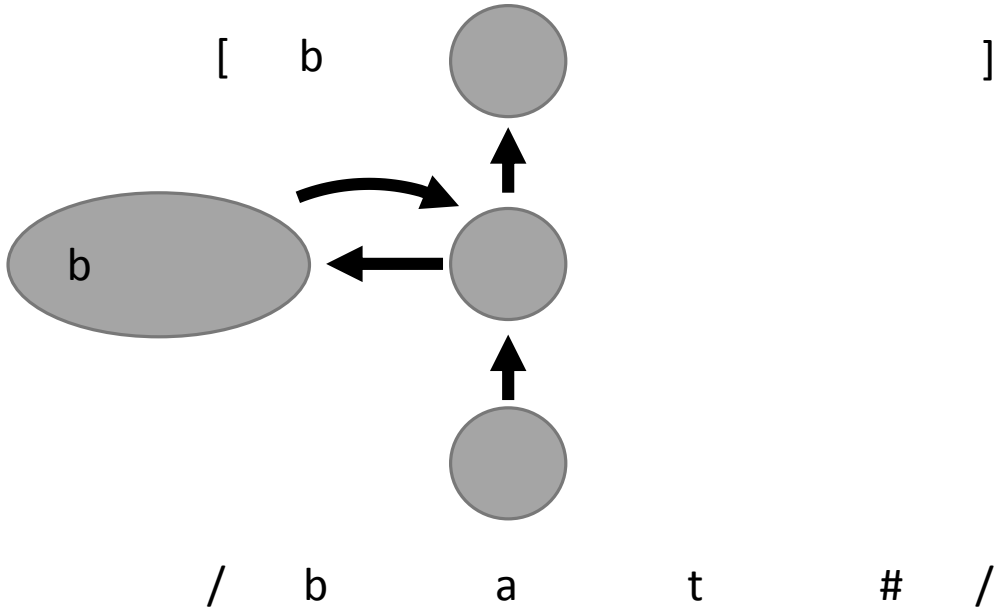
- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, segments) at a time.



/ b a t # /

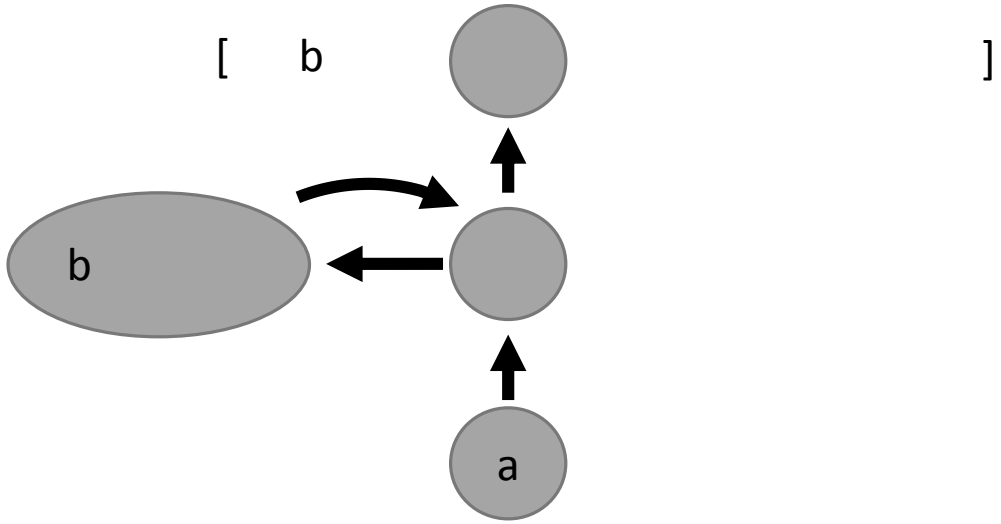
Recurrent Neural Networks

- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, segments) at a time.



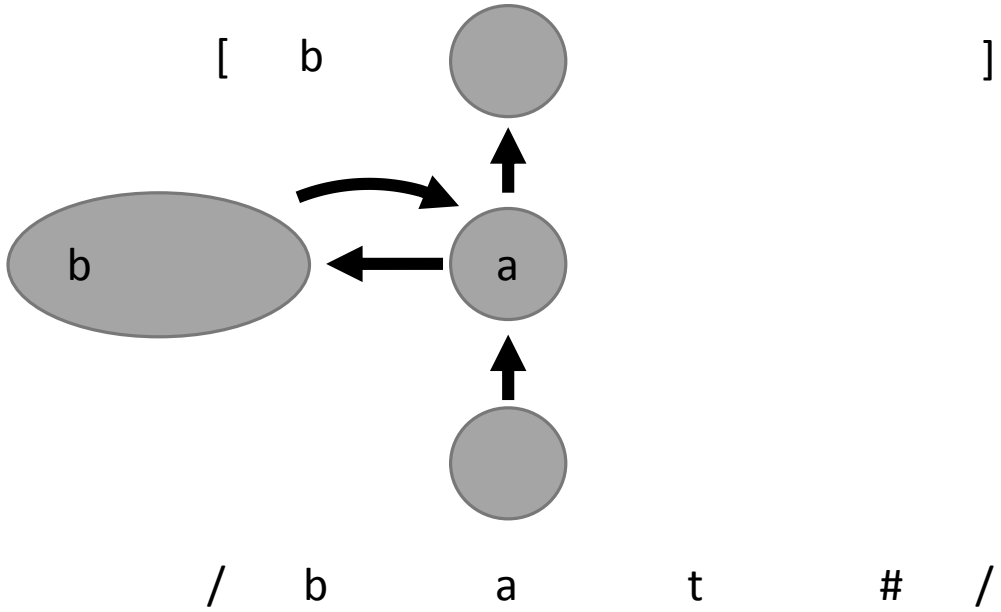
Recurrent Neural Networks

- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, segments) at a time.



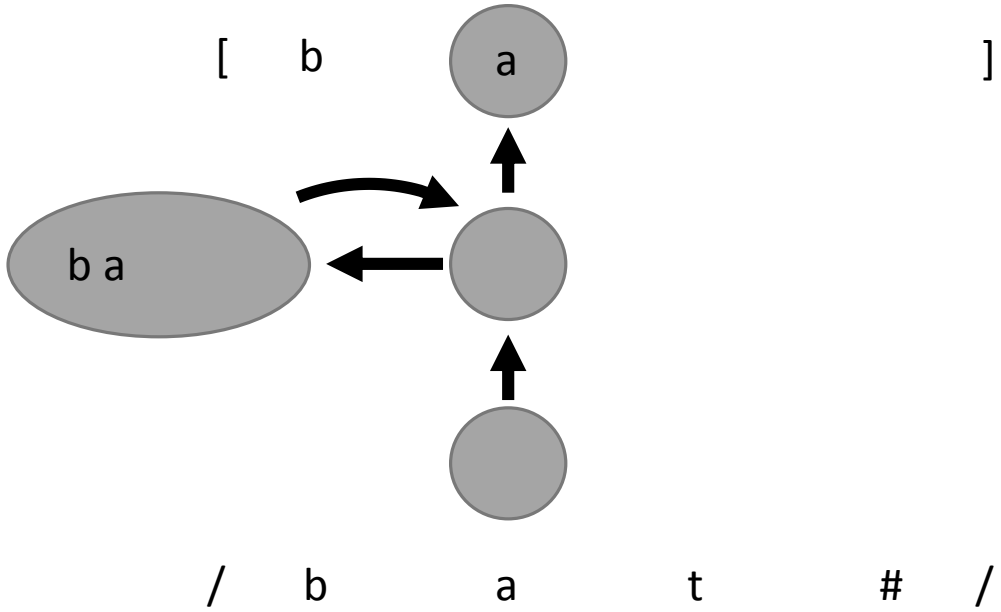
Recurrent Neural Networks

- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, segments) at a time.



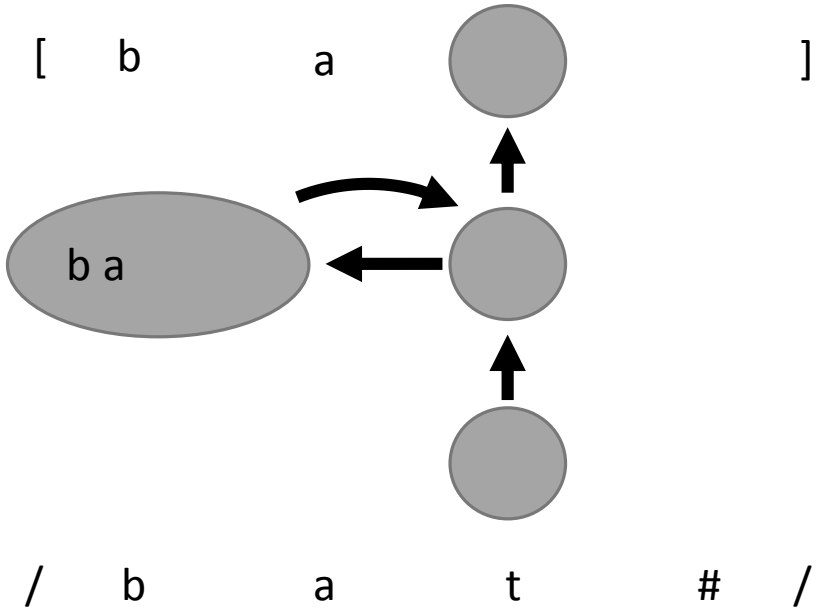
Recurrent Neural Networks

- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, segments) at a time.



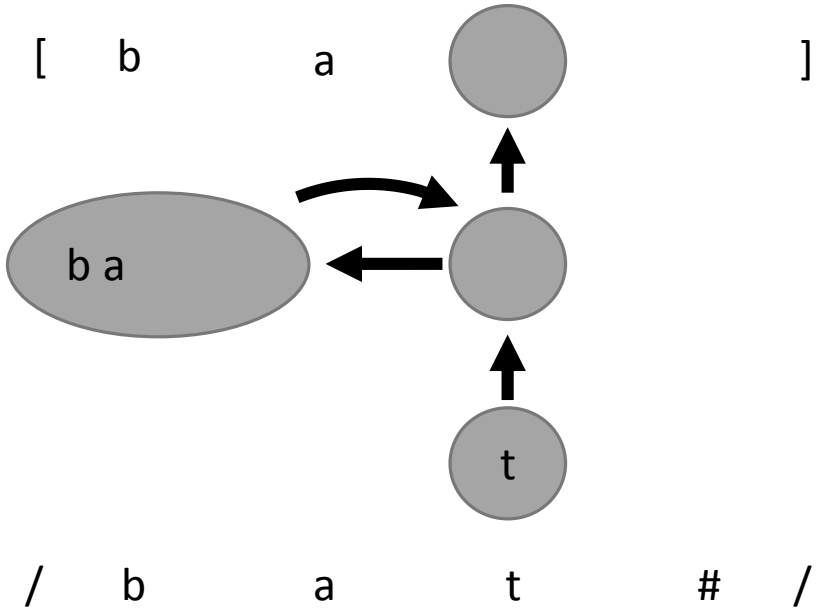
Recurrent Neural Networks

- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, segments) at a time.



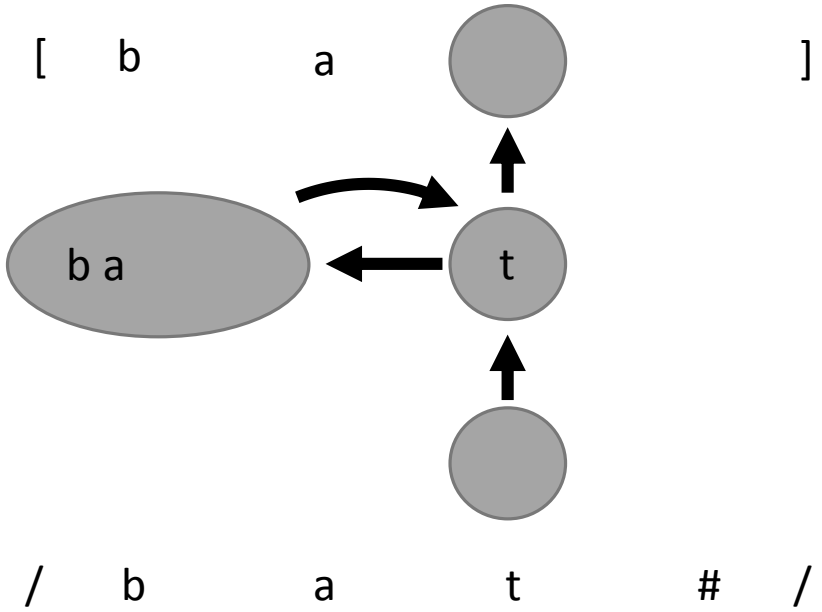
Recurrent Neural Networks

- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, segments) at a time.



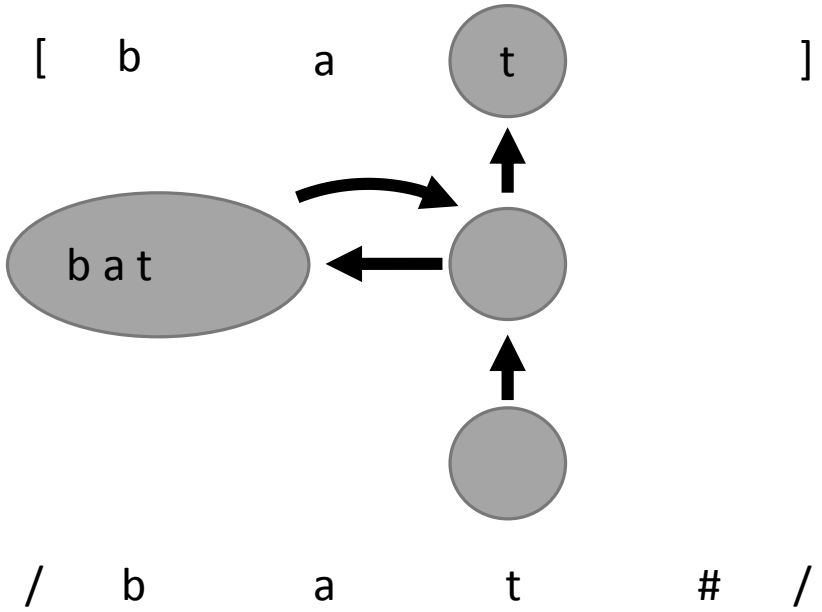
Recurrent Neural Networks

- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, segments) at a time.



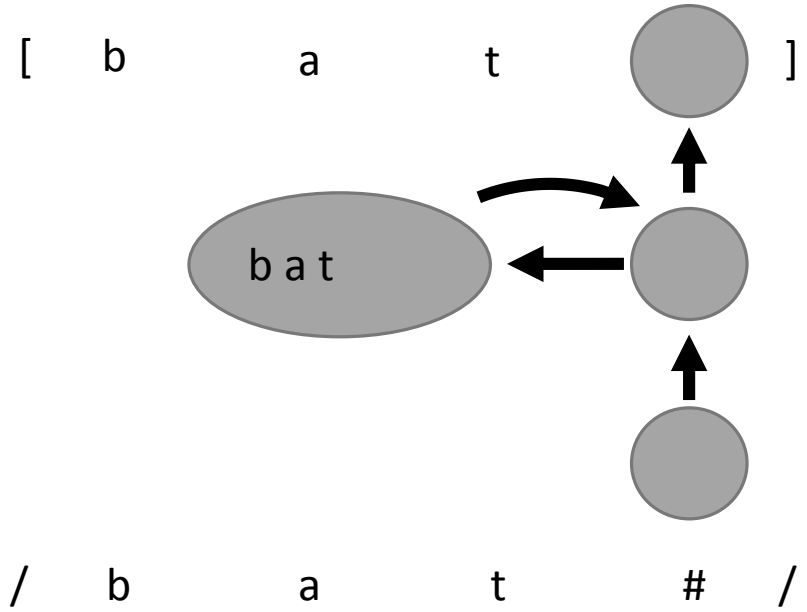
Recurrent Neural Networks

- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, segments) at a time.



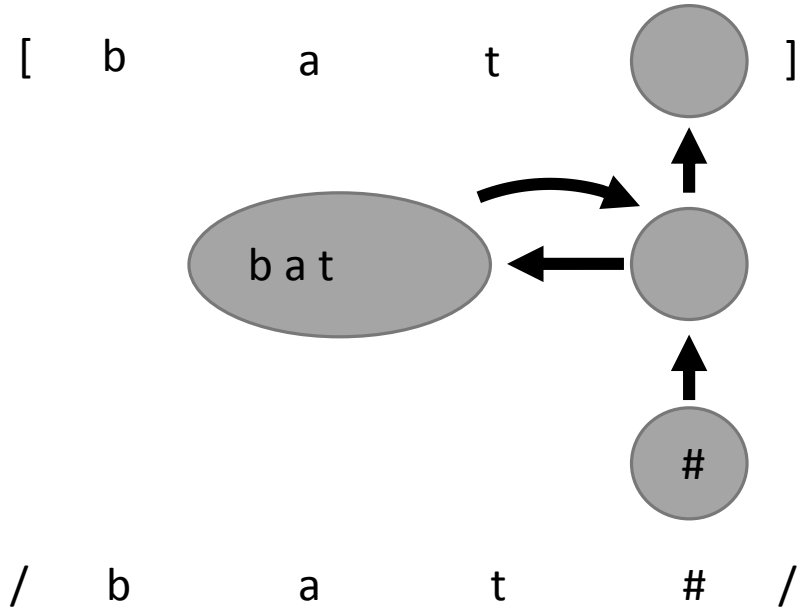
Recurrent Neural Networks

- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, segments) at a time.



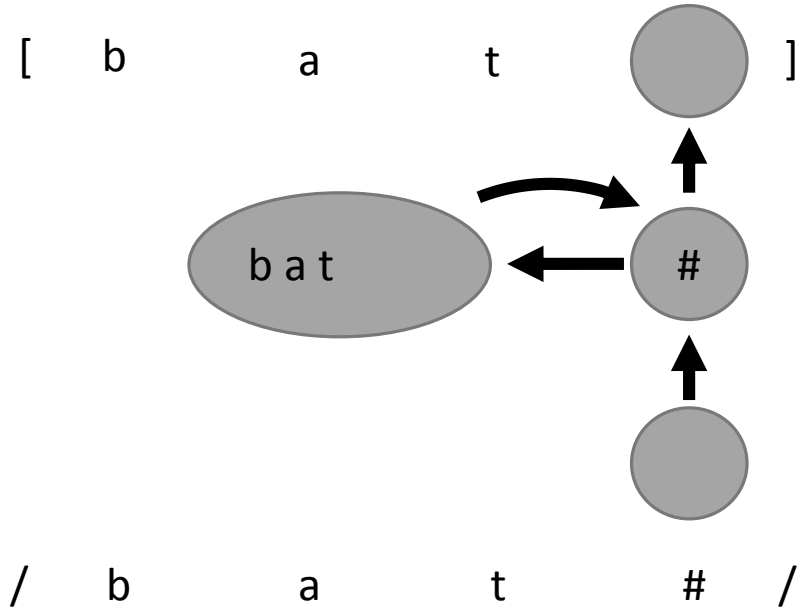
Recurrent Neural Networks

- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, segments) at a time.



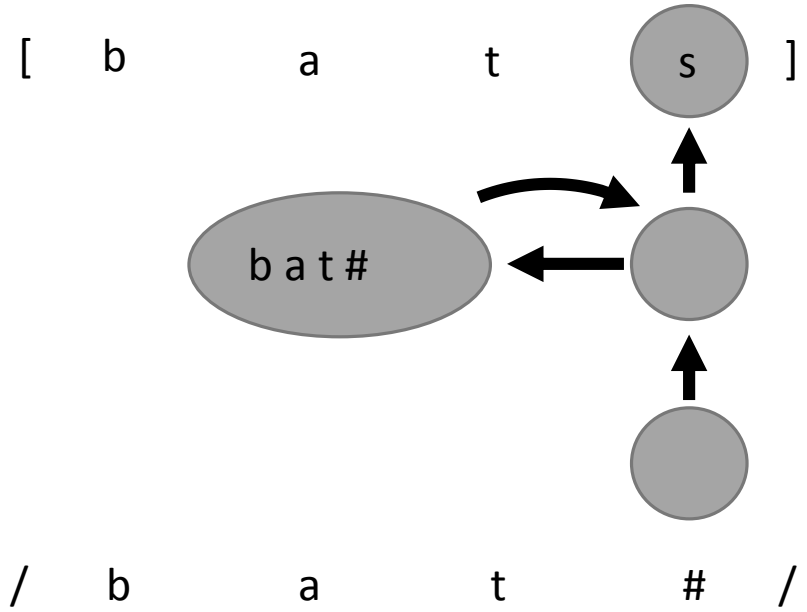
Recurrent Neural Networks

- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, segments) at a time.



Recurrent Neural Networks

- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, segments) at a time.



Recurrent Neural Networks

- Since language is something that happens over a period of time, it makes sense to work time into the architecture of our neural net.
  - This is a simplified version of what a Seq2Seq learner does.
- Recurrent Neural Nets (RNNs) do this by using a group of hidden nodes as memory and walking through a datum (in our case, a word) one moment (in our case, segments) at a time.

[b a t s]

/ b a t # /

VII. Tableaux

Bleeding (Stratal OT)

- Stratum 1

/tia/	*VV	*ti	Max	Ident
tia	W*	W*	L	
→ta			*	
tʃia	W*		L	W*
tʃa			*	W*

- Stratum 2

ta	*VV	*ti	Max	Max
→[ta]				
[tʃa]			W*	

Feeding (Stratal OT)

- Stratum 1

/tai/	*VV	*ti	Ident	Max
tai	W*		L	L
ti		W*	L	*
tʃai	W*		L	L
→tʃi			*	*

- Stratum 2

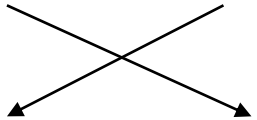
tʃi	*VV	*ti	Ident	Max
→[tʃi]				
[ti]		W*	W*	

Counterbleeding (Stratal OT)

- Stratum 1

/tia/	*ti	Max	*VV	Ident
tia	W*		*	L
ta		W*	L	L
→tʃia			*	*
tʃa		W*	L	*

- Stratum 2



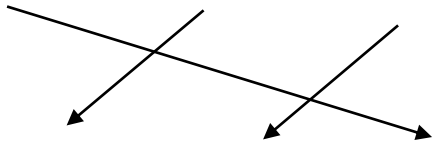
tʃia	*ti	*VV	Max	Ident
[tʃia]		W*	L	
→[tʃa]			*	
[tia]	W*	W*	L	W*
[ta]			*	W*

Counterfeeding (Stratal OT)

- Stratum 1

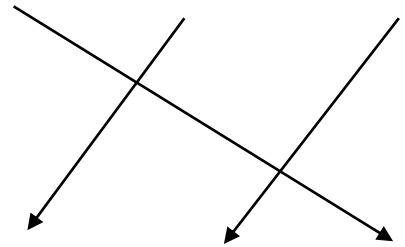
/tai/	*ti	Ident	*VV	Max
→tai			*	
ti	W*		L	W*
tʃai		W*	*	
tʃi		W*	L	W*

/ti/	*ti	Ident	*VV	Max
ti	W*			
→tʃi		W*		



- Stratum 2

tai	Ident	*VV	*ti	Max
[tai]		W*	L	L
→[ti]			*	*
[tʃai]	W*	W*	L	L
[tʃi]	W*		L	*



tʃi	Ident	*VV	*ti	Max
→[tʃi]				
[ti]	W*		W*	

Bleeding (HS with SMR constraints)

- Iteration 1

/tia/	*ti	Ident	*VV	Max	SM (*ti, *VV)	SM (*VV, *ti)
tia <>	W*		W*	L	L	L
→ta <*ti+*VV>				*	*	*
tʃia <*ti>		W*	W*	L	L	L

- Iteration 2

ta <*ti+*VV>	*ti	Ident	*VV	Max	SM (*ti, *VV)	SM (*VV, *ti)
→[ta] <*ti+*VV>					*	*
tʃa <*ti+*VV>		W*			*	*

Feeding (HS with SMR constraints)

- Iteration 1

/tai/	*VV	*ti	Ident	Max	SM (*ti, *VV)	SM (*VV, *ti)
tai <>	W*	L		L		
→ti <*VV>		*		*		
tʃai <*ti>	W*	L	W*	L		

- Iteration 2

ti <*VV>	*VV	*ti	Ident	Max	SM (*ti, *VV)	SM (*VV, *ti)
ti <*VV>		W*	L		L	
→tʃi <*VV, *ti>			*		*	

Counterbleeding (HS w/ SMR)

- Iteration 1

/tia/	*ti	SM (*ti, *VV)	Ident	*VV	Max	SM (*VV, *ti)
tia <>	W*		L	*		
ta <*ti+*VV>		W*	L	L	W*	W*
→tʃia <*ti>			*	*		

- Iteration 2

tʃia <*ti>	*ti	SM (*ti, *VV)	Ident	*VV	Max	SM (*VV, *ti)
tʃia <*ti>				W*	L	L
→tʃa <*ti, *VV>					*	*

Counterfeeding (HS w/ SMR)

- Iteration 1

/tai/	SM (*ti, *VV)	*VV	*ti	Ident	Max	SM (*VV, *ti)
tai <>		W*	L		L	
→ti <*VV>			*		*	
tʃai <*ti>		W*	L	W*	L	

- Iteration 2

ti <*VV>	SM (*ti, *VV)	*VV	*ti	Ident	Max	SM (*VV, *ti)
→[ti] <*VV>			*			
tʃi <*VV, *ti>	W*		L	W*		