A Supplementary Materials for A Neural Model of Adaptation in Reading

A.1 Language model

We used the model trained by Gulordava et al. (2018). This model was trained on 90 million words of English Wikipedia articles. It had two LSTM layers with 650 hidden units each, 650-dimensional word embeddings, a learning rate of 20, a dropout rate of 0.2 and a batch size 128, and was trained for 40 epochs (with early stopping).

A.2 Analysis of reading times

We fit linear mixed-effects models (LMEM) using the *lme4* R package (Bates et al., 2014). Our largest LMEM contained random intercepts for items and subjects, and fixed effects and by-subject random slopes for word length, sentence position, non-adaptive surprisal from the base model, and adaptive surprisal. The full LMEM formula in R notation is as follows:

> RT ∼ word_length + sentence_position + non-adaptive_surprisal + adaptive_surprisal + (1|word) + (1 + word_length + sentence_position + non-adaptive_surprisal + adaptive_surprisal |subject)

To assess whether a predictor significantly contributes to the model's fit to the data, we used a likelihood ratio test comparing an LMEM that includes that predictor with an LMEM that does not.

A.3 Fine and Jaeger (2016) simulation

We reproduce here the examples of the materials from the introduction. The critical region (where surprisal was evaluated to assess adaptation to the higher probability of reduced relative clauses) is underlined.

(1) *Ambiguous:*
The experienced soldiers warned about the dangers conducted the midnight raid.

(2) *Unambiguous:*
The experienced soldiers who were warned about the dangers conducted the midnight raid.

Our replication of their experiment compiled their 80 fillers and 40 critical items into 16 lists (item orders). Four randomized orderings were unique, four orderings had the same items in each position as the first four but with opposite conditions for each critical item, and each of those eight total lists were also presented in reverse order, producing 16 stimulus lists. Each stimulus list contained an ambiguous or an unambiguous version of each critical item but not both, and all filler sentences were present in each list.

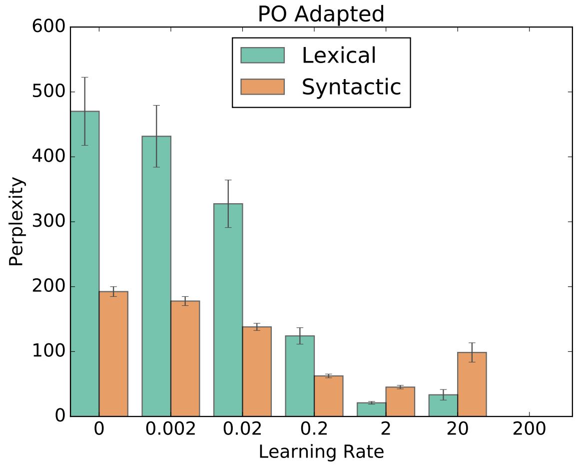

Figure 1: Learning rate influence over syntactic and lexical adaptation. A learning rate of 0 indicates the performance of the non-adaptive model; the learning rate of 200 resulted in perplexity in the billions.

A.4 Dative alternation simulation

We repeated the dative DO adaptation experiment described in the text 10 times, with different critical items and filler sentences in a randomized order for each iteration; we report averaged results and plot the means and standard deviations in our bar charts. We also conducted similar experiments with the roles of the two constructions reversed (the adaptation set included PO instead of DO sentences). The results were very similar to the DO results we report in the paper (see Figure 1). Again, the model initially assigns a lower probability to DO constructions which is the reason behind the non-adaptive model's different performance on the lexical adaptation test set (DO here, PO in the paper) compared with the syntactic adaptation test set (PO here, DO in the paper). In the PO adapted model, at the optimal learning rate, lexical adaptation was sufficient to overcome this syntactic pre-training bias.

References

Douglas Bates, Martin Maechler, Ben Bolker, and Steven Walker. 2014. *lme4: Linear mixed-effects* *models using Eigen and S4*. R package version 1.1-7.

Alex B. Fine and T. Florian Jaeger. 2016. The role of verb repetition in cumulative structural priming in comprehension. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 42(9):1362–1376.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205. Association for Computational Linguistics.