The importance of neutral baselines: Cross-structural priming ≠ Shared representation

Grusha Prasad (gprasad@colgate.edu) *Colgate University*
Aniello De Santo *University of Utah*
Shota Momma *UMass Amherst*

Introduction: Structural priming, a widely used tool to study people's sentence representations, often relies on the following logic: if a structure A is facilitated more by structure B compared to a control C, then the representations of A and B have shared components [1]. However, recent work has suggested that this logic might not hold [2]. Specifically, this work found that the priming for reduced RCs (Tab 1: A) is graded: people produced reduced RCs most frequently after other reduced RCs, less frequently after other passive RCs (Tab 1: B,C), and least frequently after control sentences with no RCs (Tab 1: D). Crucially, this work implemented an ACT-R based parser (SPAWN) and showed that these <u>graded priming effects can be derived from this parser without assuming any shared structure</u> between reduced RCs and other passive RCs in the grammar (cf. [3]). They hypothesized that this gradation was a likely result of facilitation driven by SPAWN's reanalysis mechanism, not shared structure in the grammar. To better understand the underlying cause of the gradation, we implemented a simplified version of SPAWN and added an additional no-prime control condition. The no-prime condition revealed that unlike what [2] suggested, <u>the gradation emerged not from facilitation due to reanalysis, but from an *inhibition* of reduced RCs with the control sentences.</u>

Model description: SPAWN is a fully implemented ACT-R based serial parser. Processing a word in a sentence involves retrieving a CCG style tag for the word and integrating it with the current parse state if possible, or reanalyzing otherwise. The retrieval of a tag is based on ACT-R principles influenced by several factors. However, not all of these factors are directly relevant to understanding priming, and might thus obfuscate the true mechanisms behind the process. To systematically study the cause of the priming effect, we implemented a simplified version of SPAWN that isolates three factors most relevant for priming: <u>relative frequencies of different structures, strength of prior knowledge, and strength of priming</u>. Tab 2 summarizes the differences between SPAWN and our simplified model.

Generating predictions: We used the Prime-Target set up from [2] (Tab 1), and added an additional no-prime condition. We used corpus counts [4] to set the relative frequencies and systematically varied the strength of prior knowledge (S_K), and the strength of priming (S_P) with two hyperparameters (Tab 3). We used the relative frequencies and S_K to determine the starting counts for each structure (C_S), and S_P to update this count after priming ($C_{U|P}$). To generate predictions we used these counts (C_S for no-prime and $C_{U|P}$ for each prime P) to sample CCG tags for the ambiguous verb. Then, we measured the proportion of times a tag consistent with the reduced RC parse was retrieved in the no-prime and each of the prime conditions.

Results: As expected, the priming effect was directly proportional to the prime strength, and inversely proportional to the prior strength hyperparameters. We replicated the graded pattern for more than half of the hyperparameter settings: the highest proportion of reduced RC tag was retrieved with reduced RC primes, followed by the passive RC primes, and lowest with the no RC primes. Crucially, the no-prime condition had the same or higher proportion of reduced RC tags as passive RC primes, suggesting that there was no facilitation in these conditions, unlike what [2] suggested. Instead, the gradation emerged from the no RC primes making the reduced RC primes less likely than baseline ("anti-priming"). See Fig 1.

Discussion: Our results suggest that graded priming can emerge from "anti-priming" effects because what intuitively appears to be neutral baselines may not be so. Thus, in contrast with the typical priming logic [1], graded priming alone is insufficient to infer shared structure. More broadly, this work emphasizes the need to derive priming predictions from a computational model that is explicit about its structural and processing assumptions in order to expose potentially incorrect intuitions about neutral baselines, and minimize imprecise or incorrect interpretations of priming results.

References:
[1] Branigan, Holly P & Martin J Pickering (2017). An experimental approach to linguistic representation. BBS.
[2] Prasad & Linzen (2024). SPAWNing structural priming predictions from a cognitively motivated parser. CoNLL.
[3] Harwood, William (2018). Reduced Relatives and Extended Phases: A Phase-Based Analysis of the Inflectional Restrictions on English Reduced Relative Clauses. Studia Linguistica.
[4] Roland,D.,Dick,F.,& Elman,J. L.(2007).Frequency of basic English grammatical structures: A corpus analysis. JML

Primes:
A. The cat examined by the doctor was skittish. (Passive reduced RC; RRC)
B. The cat being examined by the doctor was skittish. (Passive progressive RC; ProgRRC)
C. The cat who was examined by the doctor was skittish (Passive full RC; FRC)
D. The cat examined the doctor and was skittish (Main Verb no RC; AMV)
Target:
The defendant examined ___

Tab 1: Prime-target set up in [2]

Operation	SPAWN	Our simplified model
Get base counts	Create training data with desired distributions. Parse training sentences, update counts based on parse	Specify relative frequencies and start counts. Multiply the two.
Get base activation	Log transform counts after applying decay over time	Log-transform counts without decay
Retrieve tag given parse state	Combine base activation, lexical activation, inhibition from parse state, and noise. Pick tag with max activation. If the tag cannot combine, discard and retrieve another tag.	Combine base activation and noise. Pick tag with max activation. If the tag cannot combine, discard and retrieve another tag.
Check if retrieved t tag can combine with parse state.	Apply CCG combination rules to a fully specified grammar	Use an oracle with hard coded rules without a fully specified grammar

Tab 2: Implementational differences between SPAWN and our simplified model

Model component	How it was set
Relative frequencies	RRC: 0.012*30, FRC: 0.003*30, ProgRRC: 0.005*30, AMV: 1 Frequencies for the RCs were 30 times the frequencies in [4]. Original frequencies too low to generate any RRC responses in other conditions.
Base counts	Prior strength SK; number of sentences before experiment: 1, 10, 100 Base count = relative frequency x Prior strength
Strength of priming	SP; Number added to count after priming: 1, 5, 10 This is equivalent to the number of primes depending on grammar assumptions.

Tab 3: Description of how different model components were specified

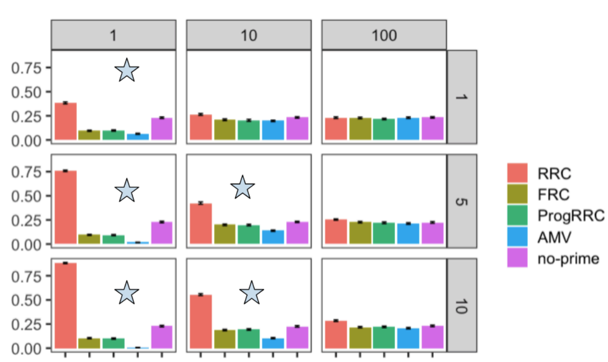

Fig 1: Predictions from our simplified model. Panels with star indicate hyperparameter settings in which the model captured the human graded priming from [2].