Different Ways to Forget: Linguistic Gates in Recurrent Neural Networks

Cristiano Chesi[1] **Veronica Bressan**[1] **Matilde Barbini**[1] **Achille Fusco**[1]
Maria Letizia Piccini Bianchessi[1] **Sofia Neri**[1] **Sarah Rossi**[1] **Tommaso Sgrizzi**[1]
[1]NeTS, IUSS Pavia
{cristiano.chesi, veronica.bressan, matilde.barbini, achille.fusco,
letizia.piccinibianchessi, sofia.neri, sarah.rossi, tommaso.sgrizzi}
@iusspavia.it

Abstract

This work explores alternative gating systems in simple Recurrent Neural Networks (RNNs) with the intent to induce linguistically motivated biases during training, ultimately affecting models' performance on the BLiMP task. Here we focus on the BabyLM 10M training corpus only (Strict-Small Track). Our experiments reveal that: (i) standard RNN variants—LSTMs and GRUs—are insufficient for properly learning the relevant set of linguistic constraints; (ii) quality and size of the training corpus have little impact on these networks since we observed comparable performance of LSTMs trained exclusively on the child-directed speech portion of the corpus; (iii) increasing the size of the embedding and hidden layers does not significantly improve performance. In contrast, specifically gated RNNs (eMG-RNNs), inspired by certain Minimalist Grammar intuitions, exhibit advantages in both training loss and BLiMP accuracy although their performance is not yet comparable to that of humans.

1 Introduction[1]

Despite their impressive performance, transformers-based architectures (Vaswani et al., 2017) provide limited insight from a theoretical linguistic perspective and tend to perform poorly when trained on small datasets, unless ad-hoc optimizations are applied (Charpentier and Samuel, 2023; Xu et al., 2024). In this paper, we focus on simple recurrent architectures to explore the effect of linguistic biases potentially induced by specific gating systems on both cross-entropy loss and performance in forced-choice tasks such as BLiMP (Warstadt et al., 2020). The goal is to preserve the role of incremental processing, which is obfuscated by the attention mechanism in transformers while retaining the self-supervised (autoregressive) training approach. Such obfuscation arises from the fact that, while human linguistic processing operates in a strictly incremental manner (Bever, 1970), the computation of gradients required to minimize model loss during LLM training must be performed in parallel for computational efficiency. This legitimate pursuit of reducing computational complexity has led, on one hand, to attention-based approaches that operate in parallel on the full input vector, composed of a fixed-length sequence of tokens, and, on the other hand, to simplifications in RNNs—such as removing any time-dependent interaction between the hidden state and the input. This last approach ultimately transfers the computational burden from the inefficient backward propagation through time (BPTT) approach to the need for additional layers (Feng et al., 2024). A relevant challenge to the Poverty of Stimulus hypothesis (Yang et al., 2017) can then be formulated in the following terms: Can a Small Language Model (SLM)—trained with a limited amount of data and under ecological exposure comparable to that of young learners—attain an adult level of linguistic competence (Chomsky, 1965)? From this perspective, linguistic competence is measured simply by the model's performance on each BLiMP subtest: a SLM will be considered *consistent*—i.e., displaying adult-like linguistic competence—if it systematically selects (> 72-80% of the times[2], sentences like "Susan

[1] Preprocessing, tokenization, models' architecture, training procedure and results are available at: https://github.com/cristianochesi/babylm-2024

[2] This is a prudential threshold obtained from the average human performance reported on BLiMP (~88%, Warstadt et al., 2020) minus 1 or 2 standard deviations (~8%).

revealed herself", which satisfy anaphoric binding (Condition A or similar generalization predicting that an anaphor like *herself* must be bound within the relevant domain, Chomsky, 1981) over the minimally different alternative "Susan revealed themselves," despite irrelevant lexical variations. We focus our analysis solely on the BLiMP minimal pair decision task to avoid complex issues related to general acceptability and coherence considerations required to interpret raw probability outputs (Lau et al., 2017).

To preserve the cognitively plausible, albeit computationally inefficient, incremental approach, we adapted Recurrent Neural Network (RNN) models (Elman, 1990) and made minimal modifications to the standard LSTM (Hochreiter and Schmidhuber, 1997) and GRU (Cho et al., 2014) architectures through gating alterations. We recorded the effects of these modifications on training loss and accuracy, and we compared the models' performance on the BLiMP task. For comparison, we also report the performance of well-studied LSTM and GRU architectures (Gulordava et al., 2018; Chowdhury and Zamparelli, 2018) after training on the 10M tokens dataset provided for the BabyLM Challenge 2024 Strict-Small Track. We hypothesized that by modifying the information flow within the network, we could provide architectural scaffolding for C-command constraints, as defined in §3 (Reinhart, 1976). We then considered two distinct computational pathways: one for preserving part of the memory content, whenever an indication that an unsatisfied dependency is present (*Move* gate), and the other for deciding whether to keep expanding the previous constituent—the "sequential phase", Bianchi and Chesi, 2014—or instantiating a nested constituent—embeddings. We then selectively simplify one pathway or the other to measure the impact of these alterations on various structural aspects. The paper is organized as follows: we first present the basic preprocessing steps adopted to clean the 10M-token corpus (§2.1). We then discuss the BLiMP dataset, focusing on the specific grammatical constraints necessary to correctly evaluate the relevant contrasts (§2.2). We conclude the introductory section by discussing the computational graphs representing standard LSTM and GRU architectures, finally speculating on the relevance of certain gating solutions from a linguistic perspective. Section 3 introduces the core linguistic intuitions we aim to model, along with attempts to rephrase these intuitions in simple, albeit potentially simplistic, combinatorial terms. Section 4 describes the basic architecture we tested, dubbed expectation-Based Minimalist Grammar Recurrent Neural Network—eMG-RNN—, loosely inspired by an unorthodox interpretation (Chesi, 2022) of Minimalist Grammars (Stabler, 2013; Chomsky et al., 2023). Section 5 presents the results of our tests, showing that the gating proposals effectively capture certain linguistic constraints but not others. Overall, the performance of eMG-RNNs is higher compared to that of LSTMs and GRUs. More importantly, unlike any LSTM and GRU architecture, eMG-RNNs consistently show biases towards one item of the minimal pairs (73% of the time for the correct item, 27% of the times for the incorrect one) in various phenomena (44% of the BLiMP filtered subtasks). We conclude with a general discussion on how different regimens have impacted these results and outline the next steps toward achieving a more precise implementation of the relevant linguistic biases that remain unresolved in the current experiments.

2 Training data, benchmarks, and RNN architectures

In this section, we present the preprocessing routines we adopted to prepare the training data for our models (§2.1). We then discuss some fundamental linguistic aspects related to the BLiMP task used to assess the linguistic performance of our models (§2.2). Finally, we introduce the standard RNN architectures—LSTM and GRU—used as starting points for our experiments (§2.3).

2.1 Corpus preprocessing

The original corpus provided with the Strict-Small Track of the BabyLM 2024 challenge consists of roughly 10M words. Six different sections are included: child-directed speech from CHILDES (MacWhinney, 2000), movie subtitles from OpenSubtitles (Lison and Tiedemann, 2016), the dialogue portion of the British National Corpus (BNC Consortium, 2007), telephone conversations from the Switchboard Dialog Act Corpus (Godfrey et al., 1992; Stolcke et al., 2000), written English from the Standardized Project Gutenberg Corpus

(Gerlach and Font-Clos, 2020), and from Simple Wikipedia (simplewiki/20221201). Because of similar preprocessing necessities, we grouped together under the label 'conversations' the BNC and Switchboard sections. Table 1 reports some details on corpus size and richness (Type-Token Ratio, TTR) before and after preprocessing.

Section	Before	After
	Tokens (TTR)	
CHILDES	1,920,655 (0.02)	1,913,959 (0.01)
SUBTITLES	2,041,868 (0.06)	2,399,780 (0.02)
CONVERSATIONS	1,079,286 (0.04)	1,211,618 (0.02)
GUTENBERG	2,539,489 (0.05)	2,895,199 (0.02)
WIKIPEDIA	1,453,539 (0.09)	1,546,763 (0.05)
ALL	9,034,837 (0.04)	9,967,319 (0.01)

Table 1: BabyLM 10M Corpus profile.

A uniform preprocessing pipeline is applied across all sections. This step includes converting text to lowercase, normalizing punctuation (e.g., adding spacing around punctuation, splitting lines after strong punctuation), removing extra spaces and line breaks, and preventing the incorrect splitting of abbreviations like *mr.* and *mrs.* by removing the dot after them. We relied solely on punctuation to segment sentences. After processing, the average word per sentence was 9 and 85% of sentences consisted of less than 75 words. Minor adjustments were made to accommodate the specific formatting characteristics of each section. These variations ensured that the preprocessing remained effective and adapted to the unique aspects of the data, while still adhering to a broadly uniform approach. For example, in the CHILDES and Switchboard sections, we removed metalinguistic information (e.g., speaker labels like *A: … B: …* or **CHI:*) and transcription symbols (e.g., *&-*, *&+*). Additionally, we made other minor adjustments specific to the corpus format, such as normalizing quotes, handling acronyms, and removing brackets. The goal of the preprocessing step was to remove any metalinguistic information and retain only the relevant phonological information (essentially pauses and rough intonation as indicated by question and exclamation marks). Obviously, removing speaker labels and converting everything to lowercase significantly undermines the model's performance on the GLUE, BLiMP Supplement, and EWoK tasks. However, as we have stated from the beginning, achieving better performance on these tasks was not our main goal.

2.2 The BLiMP dataset

The Benchmark of Linguistic Minimal Pairs for English (BLiMP, Warstadt et al., 2020) is a test set designed to assess the grammatical knowledge expressed by LLMs in English. It includes 67 groups of phenomena, each consisting of 1,000 minimal pairs of sentences that sharply contrast in grammatical acceptability. The phenomena are categorized into 12 distinct areas, such as anaphor agreement, binding, control/raising, determiner-noun agreement, ellipsis, filler-gap dependencies, irregular forms, and island effects. The pairs are generated using grammatical templates and the estimated individual human agreement with the judgments is 88.6% overall (based on judgments on 100 annotations from each paradigm). N-gram, LSTM, and Transformer language models are evaluated by assessing whether they assign a higher probability to the grammatically correct sentence in each minimal pair. To mitigate issues arising from sentence length differences (as models that sum the log probability of each token may simply penalize longer sentences), the length of the minimal pairs was kept constant. However, this approach limits the minimal contrasts that can be tested. For instance, we cannot infer from the test whether simply filling a proper gap in a wh-question influences the returned probability (e.g., "*who* do you believe X criticized $\__{who}$?" vs. "*who* do you believe X criticized Y?"). In such cases, the legitimate option adopted in BLiMP is to alternate a *wh-* item like *who* with the complementizer *that,* as in "X figured out *that* Y appreciates Z" vs. "X figured out *who* Y appreciates Z"—see §3.

2.3 Models' architecture: RNNs strike back

Although RNNs have been largely surpassed by transformers in nearly all NLP tasks, their cognitive transparency remains commendable. A recent resurgence, in the past couple of years, has also shown that both training efficiency and state-of-the-art performance can still be achieved (Feng et al., 2024; Gu and Dao, 2024). The simple idea that learning can be reduced to improving next-token prediction using word-by-word self-supervision is sufficiently ecological in the sense that it fits with the observation that children do not use adults' supervision in language acquisition (Yang et al., 2017). Importantly, research has demonstrated that pre-trained transformer-based LMs exhibit

significant differences from human performance, for instance, in correlation with reading times (Steuer et al., 2023) and in how they handle negation (Ettinger, 2020) and word order (Pham et al., 2021). More generally, the fact that transformers process all words in a sentence simultaneously (using self-attention) does not allow us to capture the incremental processing typical of human language understanding.[3] This process plays in fact a crucial role in the cognitive parsing of syntactic dependences (Frazier, 1987). Linguistic intuition often involves building up meaning incrementally, which RNNs inherently capture through their sequential processing. The LSTM architecture, for instance, centers around a sequence of gates and states that regulate information flow, possibly mirroring some relevant cognitive notion of short-term and long-term memory in language comprehension. This interpretability enables us to gain insights into how linguistic properties are represented and handled. Conversely, the (self-)attention mechanism is more opaque, involving multiple layers of attention heads that can be challenging to interpret from a linguistic perspective. We think it is then important to explore further the gating system at least in LSTM and GRU standard architectures.

2.3.1 LSTM

In all computational graphs that follow, x represents the input, h the hidden layer—or the main output—, and c an additional contextual output—if present; E represents the "embedding" consolidation—a simple linear transformation from one-hot encoded input to a lower dimensionality vector. The symbol "$\frown$" denotes the concatenation operation, while σ and *tanh* refer to the *sigmoid* and *tanh* transformations respectively. $\odot$ represents the Hadamard product and + the summation. Adopting these conventions, a standard LSTM network is described in Figure 1. One of the crucial gates in this architecture is the so-called *forget gate*, denoted as f. Due to the sigmoid transformation, when the result is multiplied ($\odot$) by the cell activation c_t, certain components in c_t are deleted, or "forgotten", whenever the f activation output values close to 0.

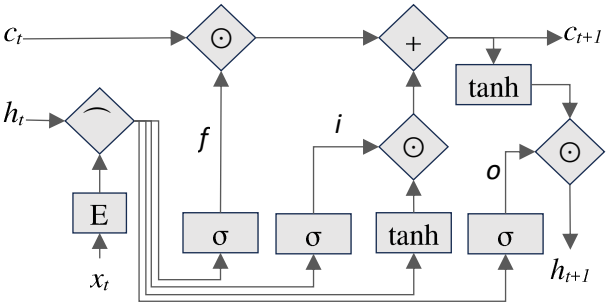

Figure 1: LSTM computational graph

In fact, the intent of the LSTM gating system was exactly to create a double pathway to process, on the one hand, the local contribution of each input component (i pathway), on the other, the long-distance contribution of c_t. Linguistically speaking, while the concept of "forgetting" seems transparent and powerful to us, both the formation of the output (o) and the contribution of the context to this output—a *tanh* transformation multiplied ($\odot$) by o—seem linguistically too unconstrained: Aside from the sigmoid transformation, a crucial decision must be made based on the simple concatenation of the hidden state and the input in both the f and o gates.

As far as the BLiMP test is concerned, the performance of the best-performing LSTM architecture—consisting of 650 embedding units (henceforth abbreviated as E650) and 650-units in each of the two hidden layers (henceforth abbreviated as H650) (Gulordava et al., 2018)—trained with 90M tokens from English Wikipedia (Warstadt et al., 2020) is reported in Table 2 below.

	LSTM	***Human***
Overall	68.9	88.6
Ana. agr	91.7	97.5
Arg. str	73.2	90
Binding	73.5	87.3
Ctrl. raising	67	83.9
D-N agr	85.4	92.2
Ellipsis	67.6	85
Filler. gap	72.5	86.9
Irregular	89.1	97
Island	42.9	84.9
Npi	51.7	88.1
Quantifiers	64.5	86.6
S-V agr	80.1	90.9

Table 2. LSTM and Human performance on BLiMP (Warstadt et al., 2020)

[3] On our limited capacity to process tokens "in parallel" one might be interested in the rapid parallel visual presentation (RPVP) task (Snell and Grainger, 2017) and on the relevant restrictions observed during this task (Fallon and Pylkkänen, 2024).

Kuncoro et al. (2018), among others, examined the impact of incorporating syntactic information into LSTM models, using syntax-sensitive dependencies like subject-verb agreement. They adapted Recurrent Neural Network Grammars (RNNGs), which utilize hierarchical phrase-structure trees, and found that while LSTMs can learn syntax-sensitive dependencies when given sufficient capacity, their accuracy declines as the number of attractors increases due to a bias toward more recent sequential information. RNNGs, which explicitly model syntactic structures through hierarchical representations, performed better than LSTMs, highlighting the importance of how syntactic structures are integrated into a model.

2.3.2 GRU

Gated recurrent units (Cho et al., 2014) can be interpreted as simplified LSTM networks that avoid storing information on an independent context output and attempt to control non-local information by means of a clever Update gate (u), as illustrated in Figure 2.

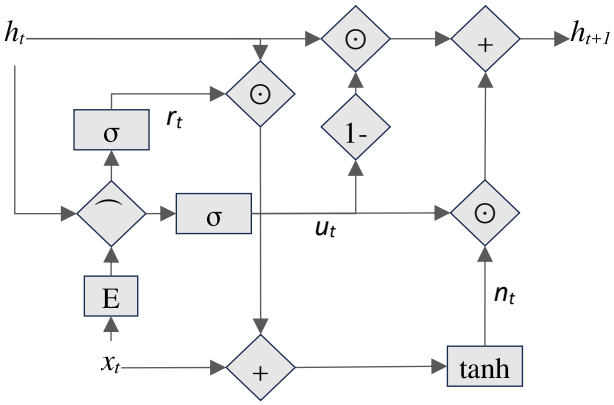

Figure 2: GRU computational graph

The output of each cell (h) is conditioned by the contribution of the update gate (u): the higher u, the greater the contribution of the previous hidden activation passed through the new gate information (n)—this is because h is multiplied (⊙) by 1-u; the lower u, the greater the contribution of the—modified—previous activation h—the input elaboration is simply multiplied (⊙) by u. The linguistic interest for this specific mechanism will be explained in the next section.

3 Core linguistic constraints and gates

According to Minimalism (Chomsky et al., 2023), *Merge* (*M*) is the fundamental structure-building operation. It is recursive, in the sense it applies to the result of other Merge operations, it is binary since it always takes two arguments, and it is local since the elements that Merge must be adjacent. Suppose *a*, *b*, *c*, *d*, and *e* are lexical items, then:

(1) $M(M(e, M(c, d)), M(a, b))) =$
$\{\{e\ \{c\ d\}\}\ \{a\ b\}\}$

That is, the structure obtained after the application of four *M* operations in this exact order—i. $M(a, b)$, ii. $M(c, d)$, iii. $M(e, \{c\ d\})$, iv. $M(\{e\ \{c\ d\}\}, \{a\ b\})$—yields a nested constituent $\{c\ d\}$ that cannot enter into any relevant structural relation with the constituent $\{a\ b\}$ (e.g., anaphor binding, as in "[$_e$ the patient$_j$ [$_c$ of [$_d$ the doctor$_i$]]] [$_a$ blames [$_b$ himself$_{*i/j}$]]"). Although this approach offers significant descriptive advantages, little attention has been given to how structure-building operations can be executed in real-time. A relatively lively debate suggests that real-time considerations may be important, both supporting behavioral evidence (Zaccarella and Friederici, 2015; Chesi and Canal, 2019) and computational predictions (Kobele et al., 2013). Especially from the language acquisition perspective, we might expect fundamental constraints that operate on structure building to induce learning biases. The two key constraints we consider here are *C(onstituent)-command* and *Locality*.

C-command is a relation that can be defined between constituents (i.e., nodes) that are merged. Adapting Reinhart's (1976) original definition to Minimalism:

(2) A node *A C-commands* a node *B iff*
i. *A* is merged with *X*, and
ii. *B* is merged within *X*

Considering the structure described in (1), *e* C-commands all other nodes, while *c* and *d* none. C-command is a fundamental property for various linguistic phenomena, such as agreement (3), gap licensing (4), and pronominal binding (5):

(3) [The friends [of John]] perform/*-s well.

(4) Joel discovered [the vase]$_i$ [that Patricia might take _$_i$]. / *Joel discovered [what$_i$ Patricia might take _$_i$ the vase].

(5) [A guy]$_i$ [that has seen [the wheelbarrow]$_j$] notices himself$_i$ /*itself$_j$.

Examples (4) and (5) are taken from BLiMP, but while (5) correctly illustrates our point—the referent *a guy* C-commands the anaphor *himself,* while *the wheelbarrow* does not C-command *itself,* despite being closer to the potential anaphor— the contrast illustrated by the minimal pair (4) is a bit misleading. The ungrammatical version in the minimal pair (4) is *Joel discovered what Patricia might take the vase.* This is an example of a "doubly filled gap": the gap position in (4) is not only filled with a DP—*the vase*—but it must also be interpreted as the legitimate argumental position in which the *wh-* item—*what*—should have been originally merged. Notice that in this contrast, no C-command violation arises. (3), on the other hand, perfectly illustrates that a closer DP *John* that does not C-command—{{the friends {of {John}}} perform}—the relevant predicate *perform* cannot agree with it.

Locality selectively restricts the span of a re-Merge (aka *Move*) operation (Rizzi, 2013). A straightforward example is illustrated by the intervention effects ("superiority effect", in the case of (6), Chomsky, 1973): a dependency between two nodes is blocked or disturbed by the presence of an intervening constituent, which is itself a potential participant in that dependency—e.g. it C-commands the gap, that is, the position where the relevant wh- item must be interpreted. Observe how ungrammaticality ensues when the *wh*-element *who* blocks the movement of the *wh*-element *what*, which is "moved" from/to its argumental position:

(6) a. *What*$_i$ could Alan discover he has run around _$_i$?
b. **What* could Alan discover *who* has run around _$_i$?

More constraints. Other contrasts are illustrated in BLiMP that adhere to C-command and Locality but also involve additional considerations and constraints that we cannot address here. These considerations and constraints are relevant to *Ellipsis*, *Control*, and *Raising* phenomena, which, despite notable attempts to describe them under a unified account, remain empirically resistant to unification.

3.1 Computational Considerations on C-command and Locality

Our core idea was to modify the gating system of a RNN to allow the network to decide whether to merge items sequentially—{*a x*}, where *a* and *x* are two tokens processed in this order <*a*, *x*>—or to instantiate a nested constituent—{*a* {*x*}}. When processing an embedded constituent, the superordinate phrasal information must be retained in memory and preserved for further merge operations that might occur once the embedded constituent initialized by *x* will be completed—{*a* {*x* …} *y*}, where *y* is the next token merged with *a*, after the closure of the constituent *x*. Moreover, any item merged within a nested phrase should be "ignored" at the superordinate level, meaning that any relevant structural relations (e.g., agreement or gap licensing) in the higher phrase should fall outside the scope of the nested items.

The RNN architecture we adopted—loosely inspired by expectation-based Minimalist Grammar formalism (Chesi, 2022)—is dubbed eMG-RNN and implements two pathways, as in standard LSTMs: one for "movement"—non-local dependencies sensitive to locality and C-command, (the *c* output in the graph in Figure 3)—, the other for "Merge" that finally affects the output *h*.

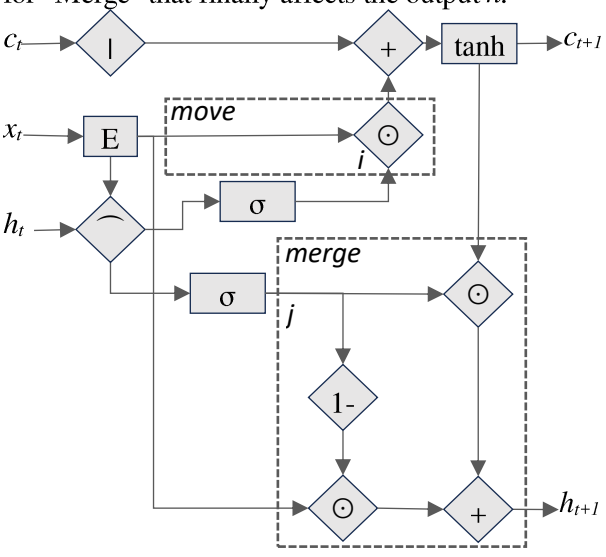

Figure 3: eMG-RNN computational graph

The gating system is slightly more complex compared to the one of LSTMs and GRUs: to contribute to the non-local activation *c*, the incoming token is first combined with the previous activation *h*, then transformed (*sigmoid*) before being applied (⊙) to the input to decide if any

relevant dependency is fully satisfied or not (*move* gate); on the other pathway, the input and the previous output *h* are combined (*merge* gate). As in the update GRUs gate, if the *merge* gate activation is robust, the incoming input will be favored (*nesting* condition); on the other hand, smaller merge activation will favor a continuation with *c* activation (*sequential* processing). The non-local *c*-activation is further transformed (*tanh*) before being passed to the next layer/output, in order to stabilize the output and model short memory decay (Lewis and Vasishth, 2005).

More precisely:

(7) $move_t=\sigma(W_{xi}x_t \frown W_{hi}h_{t-1}) \odot W_{ii}x_t$
$merge_t=\sigma(W_{xj}x_t \frown W_{hj}h_{t-1})$
$c_{t+1}=tanh(c_t + move_t)$
$h_{t+1}=(1\text{-}merge_t) \odot W_{xi}x_t + merge_t \odot c_{t+1}$

As an anonymous reviewer observed, those modifications do not guarantee at all that what we are modeling here as *move* and *merge* gates are, in fact, the Minimalist Move and Merge operations. The simple gating mechanisms employed merely assume that these operations are *unification* processes (Shieber, 1986; Chesi, 2022), where extracted features are combined and the result of unification—whether after Merge or Move—is the outcome of a simple combination of the original vectors. We propose that point-wise multiplication ($\odot$) between the concatenation of hidden $\frown$ input vectors and the input vector itself, resulting from embedding (E) is the simplest way to test this intuition. In the following experiments, we selectively modify one component (*move gate*) or the other (*merge gate*) to verify whether the reduction in accuracy resulting from these alterations aligns with the linguistic predictions that motivated this gating system.

4 Methodology

To test the new gating system, we built various RNN architectures in PyTorch (v2.4). We first implemented very simple LSTM and GRU networks, similar to the ones discussed in literature (Gulordava et al., 2018; Warstadt et al., 2020; Chowdhury and Zamparelli, 2018). Input and hidden layer(s) normalization has been evaluated and produced a slight improvement in accuracy (+.02%) and a decrease in training loss (-.2) on average when compared to unnormalized layers. Various dropout options at the input level have been tested as well but removed to reduce training loss and increase accuracy—with a dropout=.2 the average cross entropy loss increased of .6 and a 11% accuracy reduction was observed, which is coherent with the small size of the networks used. We trained these architectures with the 10M corpus for a maximum of 20 epochs—all architecture reached a plateau at worst after 12 epochs. Both symmetrical—same number of units for the embedding layer and for the hidden layers, and asymmetrical structures—lower embedding dimensions, higher number of units in the hidden layers (Chowdhury and Zamparelli, 2018) have been tested. Following Lau et al. (2017), the model output is the negative sum of the token-by-token log likelihood—cross entropy loss—, normalized by the input length. All models used a BPE tokenizer (Sennrich et al., 2016) trained on the corpus with *min_freq* set to 3 to reduce lexicon size and speed-up training (no significant improvements are observed removing this frequency constraint). The lexicon obtained consisted of 67,328 tokens. For all experiments, the maximum sequence length was 74, batch size = 64 and learning rate = 0.002. We used torch.optim.lr_scheduler with step_size=5, gamma=0.1. We also used three different data batching regimens for training. We refer to the default maximum sequence length approach as the *redundant* regimen: the corpus was divided into overlapping sequences of 74 tokens each, disregarding sentence segmentation—[[$token_1$, $token_2$, … $token_{74}$], [$token_2$, $token_3$, … $token_{75}$], …]. This produces an exposure to ~740M tokens per epoch, which is about ten times the exposure received by 7 y.o. children. We also tested two alternative regimens, which we consider more ecological. The first is the *naturalistic* regimen, which involves line-by-line batching with no sentence segmentation special tokens, or overlapping—[[$token_1$, $token_2$, … $token_n$], [$token_{n+1}$, $token_{n+2}$, … $token_m$], …], resulting in an exposure to ~10M tokens per epoch. The second is the *conversational* regimen, where batches consist of two lines of variable length from the preprocessed text with no sentence segmentation tokens, but with one line/sentence overlapping to include minimal contextual information—[[$tokenized\_sentence_1$, $tokenized\_sentence_2$], [$tokenized\_sentence_2$,

tokenized_sentence$_3$], …]. This doubles the exposure of the naturalistic regimen, while remaining within the order of magnitude of a 7-year-old's linguistic exposure. Training was performed on a High-Performance Cluster with 2 GPU nodes, each equipped with 64 CPU cores, 4 NVIDIA A100 cards with a dedicated 1GB RAM each. Each iteration required from ~1 (single-layer GRU) to ~20 hours (4-layer eMG-RNN).

4.1 Two ways of forgetting

One crucial experiment was to simplify the *move* and the *merge* gates to verify the effects of these simplifications both on training and BLiMP task performance. In the "forget nesting" condition (F-N), h_{t+1} became:

(8) $h_{t+1}=merge_t \odot c_{t+1}$

In the "forget moving" condition (F-M), the *move* gate became:

(9) $move_t=\sigma(W_{xi}x_t \frown W_{hi}h_{t-1})$

Our predictions are summarized below:

Prediction 1. If the gating system is sufficient to express C-command and Locality, all BLiMP pairs contrasting these aspects should be captured by eMG-RNN, but not by standard GRU or LSTM.

Prediction 2. Because of the sufficiently complex gating system, no improvement is expected building eMG-RNNs with multiple hidden layers.

Prediction 3. Selectively removing one gate or the other should affect performance; however, alteration of the *move* gate is expected to produce a more significant performance deterioration—this is because the simplification of the *nesting* mechanism will simply privilege sequential processing.

5 Results

Because of the low performance of GRUs (training results with 650 units for 1-, 2- or 3-layer respectively: accuracy=.3649, .3376, .3271, loss=3.1619, 3.3312, 3.4313; BLiMP supplement=.4410, .4426, .4390, filtered=.5162, .5161, .5362), we report here only the comparisons between eMG-RNNs and standard LSTMs.

Training performance. All architectures trained under the *naturalistic* and *conversational* regimen obtained low loss value (1.98 on average) and very high accuracy (90% on average) since the first epoch—plateau after 2-3 epochs. With the *redundant* regimen, more variegate results are obtained but all architectures reached ceiling performance after ten or twelve training epochs—see Figure 4 for the best performances with this last training regimen. Asymmetric architectures (E256_H1500) achieved better training performances (higher accuracy, lower $loss_{CE}$). This higher training performance is comparable with the one obtained with symmetric LSTM (E650, H650) when only the CHILDES section was used for training (child-directed speech only regimen). No significant differences have been found adding extra layers in both architectures (ceiling performance reached with 2-3 layers in LSTMs, with 1 layer in eMG-RNNs).

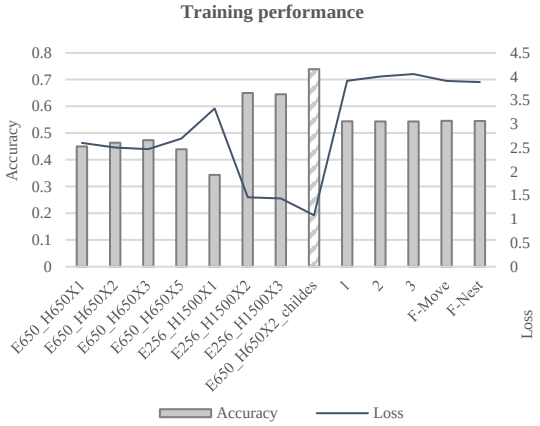

Figure 4: Best results under the redundant regimen with various LSTM architectures (labels represent architectures: E*m*_H*n*X*o* = *LSTM with m nodes as input, n nodes at the hidden layer and o layers*) eMG-RNN with *1*, *2* or *3* layers, eMG-RNN with deficient Move gate (*F-Move*) or nesting gate (*F-Nest*).

BLiMP accuracy. *Redundant* regimen produced the best results—*naturalistic* and *conversational* regimens induced a performance drop for the best architecture of ~10% and a dramatic reduction in consistency—calculated as in footnote 2. A significant lower performance is observed with LSTMs trained on 10M corpus with respect to the LSTM trained on 90M tokens Wikipedia corpus reported in the original study —Table 1. The cumulative results are reported in Table 3. The best performing LSTM architecture was the E650 H650 (Gulordava et al., 2018). Overall, the performance

of this LSTM model trained only on the CHILDES section was not significantly different (overall performance on BLiMP supplement=0.47, filtered=0.54). All eMG-RNNs, outperform LSTMs on BLiMP filtered (0.55-0.59) but the performance on BLiMP supplement is lower (0.45-0.46). Even though the cumulative performance remains low, eMG-RNN models show much more polarized preferences and very low standard errors—Appendix A. That is, accuracy on *wh*-islands and other *wh*-dependency ranges from .80 to .96, while NPI licensing goes from .02 to .11, clearly indicating a preference for the ungrammatical sentence in the pair.

		eMG-RNN				
	LSTM	**1**	**2**	**3**	**F-M**	**F-N**
Ana. agr	0.67	0.82	0.76	0.77	0.88	0.81
Arg. str	0.56	0.65	0.64	0.63	0.64	0.66
Binding	0.54	0.69	0.66	0.63	0.57	0.65
Ctrl. / Rais.	0.59	0.58	0.59	0.60	0.58	0.60
D-N agr	0.57	0.67	0.63	0.67	0.68	0.68
Ellipsis	0.41	0.24	0.30	0.21	0.42	0.39
Filler. gap	0.55	0.64	0.60	0.47	0.48	0.65
Irregular	0.54	0.58	0.69	0.60	0.60	0.58
Island	0.54	0.58	0.54	0.53	0.50	0.62
Npi	0.45	0.33	0.50	0.55	0.32	0.31
Quantifiers	0.57	0.55	0.53	0.53	0.53	0.57
S-V agr	0.50	0.52	0.52	0.52	0.55	0.53
Overall	0.54	0.58	0.58	0.57	0.55	0.59

Table 3. Aggregated performance on BLiMP. LSTM is a 2 hidden-layer network (E650-H650), eMG-RNN networks are respectively 1, 2 and 3 layers, 1-layer simplified Move (F-M) and Merge/Nesting gate (F-N)

No significant difference is observed in performance adding extra layers to the eMG-RNN models, though eMG-RNN with three layers, perform randomly on NPIs, filler-gap dependencies and islands. As far as islands are concerned, it is important to notice that the aggregate results are little informative: while 1-leyer eMG-RNN performance is random on adjunct islands, it is systematically correct on wh-islands. Lastly, simplifying the Move gate produces a significant performance drop, while even better results are obtained by "forgetting" about nesting—Merge gate simplification.

6 Discussion

Once again (Feng et al., 2024), re-exploring RNN architectures produced some noteworthy outcomes. First, we observed that with simple architectures and limited training data, classic LSTMs and GRUs are insufficient to capture meaningful linguistic generalizations. On the other hand, adopting a different gating approach, designed to support structural biases during training, leads to an improvement in forced-choice linguistic tasks. While overall accuracy remains low, this average performance conceals the interesting fact that the eMG-RNN models consistently prefer (up to 44% of the phenomena vs. 0.04% with the best performant LSTM) one option over the other—Appendix A for details. Even when the chosen option is incorrect—as in the case of NPIs—this indicates that structural biases are operative. Furthermore, as evidenced by the low standard error, lexical perturbation is marginal compared to structural inference. This point is further supported by the very low accuracy on semantic tasks, such as those required when the BLiMP supplement is performed: eMG-RNNs produce insufficient semantic generalizations. Since the goal of these experiments was to explore the transparency of simple gating options in relation to certain relevant linguistic intuitions, we conclude that our attempt is partially successful even though the gating system must be improved to capture phenomena such as control, operator-variable licensing and ellipsis. Regarding the original predictions, our experiments confirm that: (i) the gating system adopted significantly outperforms both LSTM and GRU architectures in terms of structural inferences; (ii) additional hidden layers do not improve the models' performance on structural contrasts—these architectures exhibit a very low semantic/lexical bias; and (iii) the *Move* gate appears to be much more fundamental than nesting control. This result may be consistent with the fact that, in the proposed contrasts, nesting resolution is required only in a small number of cases—something we also tend to avoid in spoken language. Lastly, the *redundant* regimen is the only one that produces effective improvement on BLiMP tasks, independent of training performance. This confirms that, despite their cognitive plausibility, these architectures do not yet challenge the Poverty of Stimulus hypothesis.

Acknowledgments

This project is partially supported by the T-GRA2L: Testing GRAdeness and GRAmmaticality in Linguistics, PRIN 2022 Next Generation EU funded Project (202223PL4N). National coordinator: CC

References

Bever, T. G. 1970. The cognitive basis for linguistic structures. *Cognition and the development of language*.

Bianchi, V. and Chesi, C. 2014. Subject islands, reconstruction, and the flow of the computation. *LINGUISTIC INQUIRY*(4):525–569.

BNC Consortium. 2007. The British National Corpus, XML Edition.

Charpentier, L. G. G. and Samuel, D. 2023. Not all layers are equally as important: Every Layer Counts BERT. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 210–224, Singapore. Association for Computational Linguistics.

Chesi, C. 2022. Expectation-based Minimalist Grammars: using the same root knowledge parsing and generation. *IJCOL*.

Chesi, C. and Canal, P. 2019. Person Features and Lexical Restrictions in Italian Clefts. *FRONTIERS IN PSYCHOLOGY*.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv:1406.1078 [cs, stat].

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*.volume 11. MIT Press, Cambridge, MA.

Chomsky, N. 1973. Conditions on transformations. In S. Anderson and P. Kiparsky, editors, *A Festschrift for Morris Halle*, pages 232–286. Holt, Rinehart and Winston, New York.

Noam Chomsky. 1981. *Lectures on government and binding: The Pisa lectures*. Walter de Gruyter.

Noam Chomsky, T. Daniel Seely, Robert C. Berwick, Sandiway Fong, M. A. C. Huybregts, Hisatsugu Kitahara, Andrew McInnerney, and Yushi Sugimoto. 2023. *Merge and the Strong Minimalist Thesis*. Cambridge University Press, 1st ed.

Chowdhury, S. A. and Zamparelli, R. 2018. RNN Simulations of Grammaticality Judgments on Long-distance Dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 133–144, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Elman, J. L. 1990. Finding Structure in Time. *Cognitive Science*, 14(2):179–211.

Ettinger, A. 2020. What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Fallon, J. and Pylkkänen, L. 2024. Language at a glance: How our brains grasp linguistic structure from parallel visual input.

Feng, L., Tung, F., Ahmed, M. O., Bengio, Y., and Hajimirsadegh, H. 2024. Were RNNs All We Needed? arXiv:2410.01201 [cs].

Frazier, L. 1987. Syntactic Processing: Evidence from Dutch. *Natural Language & Linguistic Theory*, 5(4):519–559.

Gerlach, M. and Font-Clos, F. 2020. A Standardized Project Gutenberg Corpus for Statistical Analysis of Natural Language and Quantitative Linguistics. *Entropy*, 22(1):126.

Godfrey, J. J., Holliman, E. C., and McDaniel, J. 1992. SWITCHBOARD: Telephone speech corpus for research and development. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 1:517–520.

Gu, A. and Dao, T. 2024. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv:2312.00752 [cs].

Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., and Baroni, M. 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, New Orleans, Louisiana. Association for Computational Linguistics.

Hochreiter, S. and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kobele, G. M., Gerth, S., and Hale, J. 2013. Memory Resource Allocation in Top-Down Minimalist Parsing. In Glyn Morrill and Mark-Jan Nederhof, editors, *Formal Grammar*, volume 8036 of *Lecture Notes in*

Computer Science, pages 32–51. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kuncoro, A., Dyer, C., Hale, J., Yogatama, D., Clark, S., and Blunsom, P. 2018. LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modeling Structure Makes Them Better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Melbourne, Australia. Association for Computational Linguistics.

Lau, J. H., Clark, A., and Lappin, S. 2017. Grammaticality, Acceptability, and Probability: A Probabilistic View of Linguistic Knowledge. *Cognitive Science*, 41(5):1202–1241.

Lewis, R. L. and Vasishth, S. 2005. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive science*, 29(3):375–419.

Lison, P. and Tiedemann, J. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles.

B MacWhinney. 2000. *The CHILDES Project: Tools for analyzing talk*. Mahwah, NJ: Lawrence Erlbaum Associates, Third edition.

Pham, T. M., Bui, T., Mai, L., and Nguyen, A. 2021. Out of Order: How Important Is The Sequential Order of Words in a Sentence in Natural Language Understanding Tasks?

Reinhart, T. 1976. *The syntactic domain of anaphora*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge (MA).

Rizzi, L. 2013. Locality. *Lingua*, 130:169–186.

Sennrich, R., Haddow, B., and Birch, A. 2016. Neural Machine Translation of Rare Words with Subword Units. arXiv:1508.07909 [cs].

Stuart M Shieber. 1986. *An introduction to unification-based approaches to grammar*.Lecture Notes. CSLI, Stanford, CA.

Snell, J. and Grainger, J. 2017. The sentence superiority effect revisited. *Cognition*, 168:217–221.

Stabler, E. 2013. Two Models of Minimalist, Incremental Syntactic Analysis. *Topics in Cognitive Science*, 5(3):611–633.

Steuer, J., Mosbach, M., and Klakow, D. 2023. Large GPT-like Models are Bad Babies: A Closer Look at the Relationship between Linguistic Competence and Psycholinguistic Measures. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 114–129, Singapore. Association for Computational Linguistics.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Van Ess-Dykema, C., and Meeter, M. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. 2017. Attention Is All You Need. *arXiv:1706.03762 [cs]*. arXiv: 1706.03762.

Warstadt, A., Parrish, A., Liu, H., Mohananey, A., Peng, W., Wang, S.-F., and Bowman, S. R. 2020. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Xu, W., Wu, Q., Liang, Z., Han, J., Ning, X., Shi, Y., Lin, W., and Zhang, Y. 2024. SLMRec: Empowering Small Language Models for Sequential Recommendation. arXiv:2405.17890 [cs].

Yang, C., Crain, S., Berwick, R. C., Chomsky, N., and Bolhuis, J. J. 2017. The growth of language: Universal Grammar, experience, and principles of computation. *Neuroscience & Biobehavioral Reviews*, 81:103–119.

Zaccarella, E. and Friederici, A. D. 2015. Merge in the Human Brain: A Sub-Region Based Functional Investigation in the Left Pars Opercularis. *Frontiers in Psychology*, 6.

A Appendice – Detailed BLiMP results

	LSTM		eMG-RNN							
	E650_H650X2		650x1		650x2		F-C		F-N	
categories	acc	stderr	acc	stderr	acc	stderr	acc	stderr	acc	stderr
BLiMP supplement	**0.56**	**0.01**	**0.46**	**0.01**	**0.45**	**0.01**	**0.45**	**0.01**	**0.47**	**0.01**
- hypernym	0.54	0.02	0.54	0.02	0.51	0.02	0.49	0.02	0.52	0.02
- qa congruence easy	0.42	0.06	0.33	0.06	0.31	0.06	0.39	0.06	0.36	0.06
- qa congruence tricky	0.50	0.04	0.31	0.04	0.34	0.04	0.30	0.04	0.30	0.04
- subject aux inversion	0.57	0.01	0.59	0.01	0.54	0.01	0.54	0.01	0.66	0.01
- turn taking	0.49	0.03	0.53	0.03	0.55	0.03	0.55	0.03	0.50	0.03
BLiMP filtered	**0.54**	**0.00**	**0.58**	**0.00**	**0.58**	**0.00**	**0.55**	**0.00**	**0.59**	**0.00**
- adjunct island filtered	0.45	0.02	0.49	0.02	0.35	0.02	0.43	0.02	0.51	0.02
- anaphor gender agreement filtered	0.68	0.01	0.77	0.01	0.70	0.01	0.82	0.01	0.80	0.01
- anaphor number agreement filtered	0.65	0.02	0.87	0.01	0.81	0.01	0.94	0.01	0.81	0.01
- animate subject passive filtered	0.58	0.02	0.58	0.02	0.59	0.02	0.62	0.02	0.60	0.02
- animate subject trans filtered	0.75	0.01	0.87	0.01	0.88	0.01	0.87	0.01	0.87	0.01
- causative filtered	0.45	0.02	0.54	0.02	0.52	0.02	0.50	0.02	0.59	0.02
- complex NP island filtered	0.45	0.02	0.50	0.02	0.42	0.02	0.60	0.02	0.55	0.02
- coordinate structure constraint complex left branch filtered	0.62	0.02	0.57	0.02	0.61	0.02	0.68	0.02	0.92	0.01
- coordinate structure constraint object extraction filtered	0.42	0.02	0.37	0.02	0.34	0.02	0.31	0.02	0.25	0.01
- determiner noun agreement 1 filtered	0.55	0.02	0.68	0.02	0.65	0.02	0.67	0.02	0.67	0.02
- determiner noun agreement 2 filtered	0.58	0.02	0.69	0.02	0.66	0.02	0.73	0.01	0.66	0.02
- determiner noun agreement irregular 1 filtered	0.57	0.02	0.64	0.02	0.60	0.02	0.60	0.02	0.69	0.02
- determiner noun agreement irregular 2 filtered	0.65	0.02	0.73	0.02	0.70	0.02	0.80	0.01	0.69	0.02
- determiner noun agreement with adj 2 filtered	0.54	0.02	0.64	0.02	0.58	0.02	0.66	0.02	0.61	0.02
- determiner noun agreement with adj irregular 1 filtered	0.55	0.02	0.62	0.02	0.57	0.02	0.65	0.02	0.77	0.02
- determiner noun agreement with adj irregular 2 filtered	0.56	0.02	0.70	0.02	0.67	0.02	0.75	0.01	0.68	0.02
- determiner noun agreement with adjective 1 filtered	0.55	0.02	0.65	0.02	0.60	0.02	0.60	0.02	0.64	0.02
- distractor agreement relational noun filtered	0.47	0.02	0.48	0.02	0.51	0.02	0.46	0.02	0.47	0.02
- distractor agreement relative clause filtered	0.49	0.02	0.51	0.02	0.51	0.02	0.48	0.02	0.50	0.02
- drop argument filtered	0.60	0.02	0.75	0.01	0.72	0.01	0.71	0.01	0.72	0.01
- ellipsis n bar 1 filtered	0.49	0.02	0.27	0.02	0.34	0.02	0.54	0.02	0.51	0.02
- ellipsis n bar 2 filtered	0.33	0.02	0.21	0.01	0.26	0.02	0.30	0.02	0.28	0.02
- existential there object raising filtered	0.65	0.02	0.67	0.02	0.65	0.02	0.63	0.02	0.72	0.02
- existential there quantifiers 1 filtered	0.63	0.02	0.94	0.01	0.90	0.01	0.90	0.01	0.97	0.01
- existential there quantifiers 2 filtered	0.81	0.01	0.30	0.02	0.06	0.01	0.57	0.02	0.43	0.02
- existential there subject raising filtered	0.62	0.02	0.56	0.02	0.66	0.02	0.66	0.02	0.60	0.02
- expletive it object raising filtered	0.62	0.02	0.58	0.02	0.56	0.02	0.57	0.02	0.57	0.02
- inchoative filtered	0.41	0.02	0.42	0.02	0.40	0.02	0.47	0.02	0.43	0.02
- intransitive filtered	0.42	0.02	0.62	0.02	0.62	0.02	0.64	0.02	0.65	0.02
- irregular past participle adjectives filtered	0.61	0.02	0.68	0.02	0.76	0.01	0.53	0.02	0.77	0.01
- irregular past participle verbs filtered	0.46	0.02	0.49	0.02	0.63	0.02	0.68	0.02	0.38	0.02
- irregular plural subject verb agreement 1 filtered	0.52	0.02	0.59	0.02	0.54	0.02	0.64	0.02	0.61	0.02
- irregular plural subject verb agreement 2 filtered	0.51	0.02	0.57	0.02	0.58	0.02	0.59	0.02	0.57	0.02
- left branch island echo question filtered	0.82	0.01	0.46	0.02	0.33	0.02	0.42	0.02	0.46	0.02
- left branch island simple question filtered	0.57	0.02	0.66	0.02	0.69	0.02	0.63	0.02	0.89	0.01
- matrix question npi licensor present filtered	0.24	0.01	0.48	0.02	0.69	0.02	0.77	0.01	0.07	0.01
- npi present 1 filtered	0.41	0.02	0.11	0.01	0.27	0.01	0.13	0.01	0.25	0.01
- npi present 2 filtered	0.41	0.02	0.10	0.01	0.26	0.01	0.12	0.01	0.23	0.01
- only npi licensor present filtered	0.58	0.02	0.37	0.02	0.71	0.02	0.00	0.00	0.72	0.02
- only npi scope filtered	0.36	0.02	0.02	0.01	0.01	0.00	0.07	0.01	0.31	0.02
- passive 1 filtered	0.62	0.02	0.75	0.02	0.75	0.01	0.72	0.02	0.76	0.01
- passive 2 filtered	0.66	0.02	0.76	0.01	0.75	0.01	0.70	0.02	0.75	0.01
- principle A c command filtered	0.41	0.02	0.63	0.02	0.54	0.02	0.62	0.02	0.54	0.02
- principle A case 1 filtered	0.73	0.01	1.00	0.00	1.00	0.00	0.59	0.02	0.92	0.01
- principle A case 2 filtered	0.50	0.02	0.62	0.02	0.63	0.02	0.60	0.02	0.72	0.01
- principle A domain 1 filtered	0.53	0.02	0.69	0.02	0.49	0.02	0.45	0.02	0.67	0.02
- principle A domain 2 filtered	0.53	0.02	0.50	0.02	0.55	0.02	0.51	0.02	0.47	0.02
- principle A domain 3 filtered	0.54	0.02	0.54	0.02	0.55	0.02	0.51	0.02	0.54	0.02
- principle A reconstruction filtered	0.53	0.02	0.88	0.01	0.88	0.01	0.72	0.01	0.73	0.01
- regular plural subject verb agreement 1 filtered	0.49	0.02	0.53	0.02	0.52	0.02	0.68	0.02	0.57	0.02
- regular plural subject verb agreement 2 filtered	0.49	0.02	0.44	0.02	0.47	0.02	0.47	0.02	0.49	0.02
- sentential negation npi licensor present filtered	0.62	0.02	0.66	0.02	0.70	0.02	0.57	0.02	0.38	0.02
- sentential negation npi scope filtered	0.54	0.02	0.60	0.02	0.84	0.01	0.61	0.02	0.25	0.01
- sentential subject island filtered	0.43	0.02	0.70	0.01	0.71	0.01	0.55	0.02	0.58	0.02
- superlative quantifiers 1 filtered	0.36	0.02	0.51	0.02	0.51	0.02	0.51	0.02	0.51	0.02
- superlative quantifiers 2 filtered	0.47	0.02	0.47	0.02	0.64	0.02	0.15	0.01	0.37	0.02
- tough vs raising 1 filtered	0.37	0.02	0.37	0.02	0.30	0.01	0.36	0.02	0.39	0.02
- tough vs raising 2 filtered	0.67	0.02	0.69	0.02	0.76	0.01	0.67	0.02	0.69	0.02
- transitive filtered	0.59	0.02	0.54	0.02	0.55	0.02	0.52	0.02	0.55	0.02
- wh island filtered	0.58	0.02	0.89	0.01	0.86	0.01	0.42	0.02	0.83	0.01
- wh questions object gap filtered	0.61	0.02	0.80	0.01	0.77	0.01	0.38	0.02	0.85	0.01
- wh questions subject gap filtered	0.63	0.02	0.82	0.01	0.80	0.01	0.20	0.01	0.81	0.01
- wh questions subject gap long distance filtered	0.68	0.02	0.82	0.01	0.69	0.02	0.60	0.02	0.89	0.01
- wh vs that no gap filtered	0.59	0.02	0.95	0.01	0.88	0.01	0.55	0.02	0.97	0.01
- wh vs that no gap long distance filtered	0.65	0.02	0.96	0.01	0.85	0.01	0.59	0.02	0.97	0.01
- wh vs that with gap filtered	0.38	0.02	0.07	0.01	0.08	0.01	0.58	0.02	0.03	0.01
- wh vs that with gap long distance filtered	0.35	0.02	0.07	0.01	0.16	0.01	0.44	0.02	0.02	0.00