Supplementary Information for: Modeling word and morpheme order in natural language as an efficient tradeoff of memory and surprisal

Michael Hahn, Judith Degen, Richard Futrell

September 15, 2020

Contents

1 Formal Analysis and Proofs

In this section, we prove the Information Locality Bound Theorem and related theoretical results referenced in the main paper.

1.1 Mathematical Assumptions

We first make explicit how we formalize language processing for proving the theorem. This is a formally fully rigorous statement of the model described in main paper (Section 'An information-theoretic model of online language comprehension').

1.1.1 Ingredient 1: Language as a Stationary Stochastic Process

We represent language as a stochastic process of words $W = \dots w_{-2}w_{-1}w_0w_1w_2\dots$, extending indefinitely both into the past and into the future (Doob, 1953). The symbols w_t belong to a common set, representing the words or morphemes of the language. Formally, a stochastic process is a probability distribution over infinite sequences $\dots w_{-2}w_{-1}w_0w_1w_2\dots$ (Doob, 1953). As t runs over the set of integers $\mathbb{Z}$, it will sometimes be convenient to write such an infinite sequence as $(w_t)_{t\in\mathbb{Z}}$. This distribution gives rise to probability distributions over finite subsequences

$$P(w_t, \dots, w_{t+T}) \tag{1}$$

for integers t, T, and to conditional probabilities

$$P(w_t | w_{t-T}, \dots, w_{t-1}) \tag{2}$$

Infinite Length We assume that the process W extends infinitely into both past and future, whereas real words, sentences, and conversations are finite. This is not a contradiction: In Studies 1–3, we model W as a sequence of independent sentences or words, separated with a special "end-of-sentence" symbol. Modeling W as such an infinite sequence of finite sentences provides a way to formalize the memory-surprisal tradeoff in a way independent of the time point t.

Stationarity We make the assumption that the process W is *stationary* (Doob, 1953). This means that the joint distribution of different symbols depends only on their *relative positions*, not their *absolute positions*. Formally, this means that joint probabilities do not change when shifting all observations by a constant number Δ of time steps. That is, for any integers t, Δ, and $T > 0$:

$$P(w_t, \dots, w_{t+T}) = P(w_{t+\Delta}, \dots, w_{t+T+\Delta}) \tag{3}$$

Informally, this says that the process has no 'internal clock', and that the statistical rules of the language do not change over time at the timescale we are interested in. In reality, the statistical rules of language

do change: They change as language changes over generations, and they also change between different situations – e.g., depending on the interlocutor at a given point in time. However, we are interested in memory needs in the processing of *individual sentences* or *individual words*, at a timescale of seconds or minutes. At this level, the statistical regularities of language do not change, making stationarity a reasonable modeling assumption.

The choice to model language as a stationary stochastic process is common to information-theoretic studies of text, including studies of entropy rate (Shannon, 1951; Bentz et al., 2017; Takahashi and Tanaka-Ishii, 2018), excess entropy (Debowski, 2011; Hahn and Futrell, 2019), and mutual information (Ebeling and Pöschel, 1994; Lin and Tegmark, 2017).

1.1.2 Ingredient 2: Postulates about Memory and Processing

The second ingredient consists of the three postulates about memory and processing described in the main paper. We repeat these here for reference:

1. Comprehension Postulate 1 (Incremental memory). At time t, the listener has an incremental **memory state** m_t that contains her stored information about previous words. The memory state is given by a **memory encoding function** M such that $m_t = M(w_{t-1}, m_{t-1})$.

2. Comprehension Postulate 2 (Incremental prediction). The listener has a subjective probability distribution at time t over the next word w_t as a function of the memory state m_t. This probability distribution is denoted $P(w_t|m_t)$.

3. Comprehension Postulate 3 (Linking hypothesis). Processing a word w_t incurs difficulty proportional to the **surprisal** of w_t given the memory state m_t:

$$\text{Difficulty} \propto -\log P(w_t|m_t). \tag{4}$$

We extend the assumption of stationarity explained above to the memory state m_t, modeling the pair $(w_t, m_t)_{t\in\mathbb{Z}}$ as a stationary process. Formally, this means that, for any integers t, Δ, and $T > 0$:

$$P((w_t, m_t), \ldots, (w_{t+T}, m_{t+T})) = P((w_{t+\Delta}, m_{t+\Delta}), \ldots, (w_{t+T+\Delta}, m_{t+T+\Delta})) \tag{5}$$

This means that the listener's memory state only depends on the relative temporal position of past observed symbols, not on any absolute time scale. This prevents situations where the listener's memory state keeps track of some absolute notion of time (e.g., counting whether t is even or odd) even though the statistical regularities of the input $(w_t)_{t\in\mathbb{Z}}$ are independent of time.

This assumption entails that average surprisal

$$S_M \equiv H[w_t|m_t]. \tag{6}$$

and memory cost

$$H_M \equiv \mathrm{H}[m_t] \tag{7}$$

are independent of t, as these terms only depend on the joint distribution of (w_t, m_t), which is independent of t.

1.1.3 Ingredient 3: No Mindreading

Our postulates so far do not rule out that the listener has access to information that was never revealed during past interaction. That is, they permit situations where m_t maintains some information that is not contained in the past inputs $w_{<t} = (\dots, w_{t-2}, w_{t-1})$, but is informative about future input $w_{\geq t} = (w_t, w_{t+1}, w_{t+2}, \dots)$. Such a situation would correspond to a listener 'mindreading' the speaker's intentions. We exclude this by explicitly stating that the listener has no access to information about the future beyond what is contained in the past. We formalize this as saying that the memory state is independent of future observations, conditional on the past:

$$m_t \perp w_{\geq t} | w_{<t} \tag{8}$$

Remarkably, the Information Locality Theorem can be proved even without this assumption. However, this assumption is necessary in order to prove that $S_M \geq S_\infty$ even for very large memory capacities, i.e., that imperfect memory can never lead to lower average surprisal than the entropy rate. Such a situation could only be achieved if the listener somehow 'read the speaker's mind'.

There are no further assumptions about the memory architecture and the nature of its computations.

1.2 Proof of the Theorem

Here, we prove the Information Locality Bound Theorem (Theorem 2 in the main paper) based on the assumptions described in the previous section. Recall that S_M and S_∞ are given by

$$S_M \equiv \mathrm{H}[w_t | m_t] \tag{9}$$

$$S_\infty \equiv \mathrm{H}[w_t | w_{<t}] \tag{10}$$

We restate the theorem:

Theorem 1. *Let T be any positive integer ($T \in \{1, 2, 3, ...\}$), and consider a listener using at most*

$$H_M \leq \sum_{t=1}^{T} t I_t \tag{11}$$

bits of memory on average. Then this listener will incur surprisal at least

$$S_M \geq S_\infty + \sum_{t>T} I_t \tag{12}$$

on average.

Proof. The difference between the listener's average surprisal S_M and optimal surprisal S_∞ is

$$S_M - S_\infty = \mathrm{H}[w_t | m_t] - \mathrm{H}[w_t | w_{<t}]. \tag{13}$$

Because the process $(w_t, m_t)_{t \in \mathbb{Z}}$ is stationary, we can, for any positive integer T, rewrite this expression as

$$\mathrm{H}[w_t | m_t] - \mathrm{H}[w_t | w_{<t}] = \frac{1}{T} \sum_{t'=1}^{T} \left(\mathrm{H}[w_{t'} | m_{t'}] - \mathrm{H}[w_{t'} | w_{<t'}] \right) \tag{14}$$

Due to Processing Postulate 1, we have

$$m_t = M(m_{t-1}, w_{t-1}) = M(M(m_{t-2}, w_{t-2}), w_{t-1}) = M(M(M(m_{t-3}, w_{t-3}), w_{t-2}), w_{t-1}) = \dots, \tag{15}$$

and therefore the Data Processing Inequality (Cover and Thomas, 2006) entails the following inequality for every positive integer t:

$$\mathrm{H}[w_t|m_t] \geq \mathrm{H}[w_t|w_{1\ldots t-1}, m_1]. \tag{16}$$

Plugging this inequality into Equation 14 above, we get an expression in terms of the difference in mutual information between a block of words and a memory representation, and a block of words and the true past:

$$\begin{aligned}
\mathrm{H}[w_t|m_t] - \mathrm{H}[w_t|w_{<t}] &\geq \frac{1}{T}\sum_{t=1}^{T}(\mathrm{H}[w_t|w_{1\ldots t-1}, m_1] - \mathrm{H}[w_t|w_{1\ldots t-1}, w_{\leq 0}]) && (17)\\
&= \frac{1}{T}\left(\mathrm{H}[w_{1\ldots T}|m_1] - \mathrm{H}[w_{1\ldots T}|w_{\leq 0}]\right) && (18)\\
&= \frac{1}{T}\left(I[w_{1\ldots T} : w_{\leq 0}] - I[w_{1\ldots T} : m_1]\right). && (19)
\end{aligned}$$

The first term $\mathrm{I}[w_{1\ldots T} : w_{\leq 0}]$ can be rewritten in terms of I_t using the chain rule of mutual information (Cover and Thomas, 2006):

$$\mathrm{I}[w_{1\ldots T} : w_{\leq 0}] = \sum_{i=1}^{T}\sum_{j=-1}^{-\infty}\mathrm{I}[w_i : w_j|w_{j+1}\ldots w_{i-1}] = \sum_{t=1}^{T} tI_t + T\sum_{t>T} I_t. \tag{20}$$

Therefore

$$\mathrm{H}[w_t|m_t] - \mathrm{H}[w_t|w_{<t}] \geq \frac{1}{T}\left(\sum_{t=1}^{T} tI_t + T\sum_{t>T} I_t - \mathrm{I}[w_{1\ldots T} : m_1]\right). \tag{21}$$

The term $I[w_{1\ldots T} : m_1]$ is at most $\mathrm{H}[m_1]$, which is at most $\sum_{t=1}^{T} tI_t$ by assumption. Thus, (21) implies the following:

$$\mathrm{H}[w_t|m_t] - \mathrm{H}[w_t|w_{<t}] \geq \frac{1}{T}\left(\sum_{t=1}^{T} tI_t + T\sum_{t>T} I_t - \sum_{t=1}^{T} tI_t\right) = \sum_{t>T} I_t \tag{22}$$

Rearranging yields

$$\mathrm{H}[w_t|m_t] \geq \mathrm{H}[w_t|w_{<t}] + \sum_{t>T} I_t \tag{23}$$

as claimed. □

Mutual Information as Memory Cost We model the cost of holding information memory by the entropy $H_M := \mathrm{H}[m]$. Another natural choice is the mutual information between m_t and the past, $I_M := \mathrm{I}[m_t : w_{<t}]$ (Still, 2014). Our results continue to hold for that choice: Theorem 1 remains true when replacing H_M by I_M. In the proof of the theorem, the definition of H_M enters the argument in Equation 22 through the inequality $I[w_{1\ldots T} : m_1] \leq \mathrm{H}[m_1] = H_M$. The analogous inequality for I_M remains true: $I[w_{1\ldots T} : m_1] \leq I[m_1 : w_{<1}]$ holds due to the 'No Mindreading' postulate and the stationarity of the process.

1.3 Memory-Surprisal Tradeoff in a Model with Memory Retrieval

Here we show that our information-theoretic analysis is compatible with models placing the main bottleneck in the difficulty of retrieval (McElree, 2000; Lewis and Vasishth, 2005; Nicenboim and Vasishth, 2018; Vasishth et al., 2019). We extend our model of memory in incremental prediction to capture key aspects of the models described by Lewis and Vasishth (2005); Nicenboim and Vasishth (2018); Vasishth et al. (2019).

The ACT-R model of Lewis and Vasishth (2005) assumes a small working memory consisting of *buffers* and a *control state*, which together hold a small and fixed number of individual *chunks*. It also assumes a large short-term memory that contains an unbounded number of chunks. This large memory store is accessed via *cue-based retrieval*: a query is constructed based on the current state of the buffers and the control state; a chunk that matches this query is then selected from the memory storage and placed into one of the buffers.

Formal Model We extend our information-theoretic analysis by considering a model that maintains both a small working memory m_t—corresponding to the buffers and the control state—and an unlimited short-term memory s_t. When processing a word w_t, there is some amount of communication between m_t and s_t, corresponding to retrieval operations. We model this using a variable r_t representing the information that is retrieved from s_t. In our formalization, r_t reflects the totality of all retrieval operations that are made during the processing of w_{t-1}; they happen after w_{t-1} has been observed but before w_t has.

The working memory state is determined not just by the input w_t and the previous working memory state m_{t-1}, but also by the retrieved information:

$$m_t = f(w_t, m_{t-1}, r_t) \tag{24}$$

The retrieval operation is jointly determined by working memory, short-term memory, and the previous word:

$$r_t = g(w_{t-1}, m_{t-1}, s_{t-1}) \tag{25}$$

Finally, the short-term memory can incorporate any—possibly all—information from the last word and the working memory:

$$s_t = h(w_t, m_t, s_{t-1}) \tag{26}$$

While s_t is unconstrained, there are constraints on the capacity of working memory $\mathrm{H}[m_t]$ and the amount of retrieved information $\mathrm{H}[r_t]$. Placing a bound on $\mathrm{H}[m_t]$ reflects the fact that the buffers can only hold a small and fixed number of chunks (Lewis and Vasishth, 2005).

Predictions are made based on working memory m_{t-1} and retrieved information r_t (but not the short-term memory s_t), incurring average surprisal

$$S := H[w_t | m_{t-1}, r_t]. \tag{27}$$

In line with the mathematical postulates in Section 1.1, we assume that $(w_t, m_t, r_t, s_t)_{t \in \mathbb{Z}}$ is stationary as a stochastic process.

Cost of Retrieval In the model of Lewis and Vasishth (2005), the time it takes to process a word is determined primarily by the time spent retrieving chunks, which is determined by the number of retrieval operations and the time it takes to complete each retrieval operation. If the information content of each chunk is bounded, then a bound on $H[r_t]$ corresponds to a bound on the number of retrieval operations.

In the model of Lewis and Vasishth (2005), a retrieval operation takes longer if more chunks are similar to the retrieval cue, whereas, in the direct-access model (McElree, 2000; Nicenboim and Vasishth, 2018;

Vasishth et al., 2019), retrieval operations take a constant amount of time. There is no direct counterpart to differences in retrieval times and similarity-based inhibition as in the activation-based model in our formalization. Our formalization thus more closely matches the direct-access model, though it might be possible to incorporate aspects of the activation-based model in our formalization.

Role of Surprisal The ACT-R model of Lewis and Vasishth (2005) does not have an explicit surprisal cost. Instead, surprisal effects are interpreted as arising because, in less constraining contexts, the parser is more likely to make decisions that then turn out to be incorrect, leading to additional correcting steps. We view this as an algorithmic-level implementation of a surprisal cost. If the word w_t is unexpected given the current state of the working memory—i.e., buffers and control states—then their current state must provide insufficient information to constrain the actual syntactic state of the sentence, meaning that the parsing steps made to integrate w_t are likely to include more backtracking and correction steps. Thus, we argue that cue-based retrieval models predict that the surprisal $-\log P(w_t|m_{t-1}, r_t)$ will be part of the cost of processing word w_t.

Theoretical Result We now show an extension of our theoretical result in the setting of the retrieval-based model described above.

Theorem 2. *Let $0 < S \leq T$ be positive integers such that the average working memory cost $\mathrm{H}[m_t]$ is bounded as*

$$\mathrm{H}[m_t] \leq \sum_{t=1}^{T} t I_t \tag{28}$$

and the average amount of retrieved information is bounded as

$$\mathrm{H}[r_t] \leq \sum_{t=T+1}^{S} I_t. \tag{29}$$

Then the surprisal cost is lower-bounded as

$$\mathrm{H}[w_t|m_{t-1}, r_t] \geq \mathrm{H}[w_t|w_{<t}] + \sum_{t>S} I_t. \tag{30}$$

Proof. The proof is a generalization of the proof in Section 1.2. For any positive integer t, the memory state m_t is determined by $w_{1\ldots t}, m_0, r_0, \ldots, r_t$. Therefore, the Data Processing Inequality entails:

$$\mathrm{H}[w_t|m_{t-1}, r_t] \geq \mathrm{H}[w_t|w_{1\ldots t}, m_0, r_0, \ldots, r_t]. \tag{31}$$

As in (17), this leads to

$$\begin{aligned}
\mathrm{H}[w_t|m_{t-1}, r_t] - \mathrm{H}[w_t|w_{<t}] &\geq \frac{1}{T}\sum_{t=1}^{T}\left(\mathrm{H}[w_t|w_{1\ldots t}, m_0, r_0, \ldots, r_t] - \mathrm{H}[w_t|w_{1\ldots t-1}, w_{\leq 0}]\right) && (32)\\
&\geq \frac{1}{T}\left(\mathrm{H}[w_{1\ldots T}|m_0, r_0, \ldots, r_T] - \mathrm{H}[w_{1\ldots T}|w_{\leq 0}]\right) && (33)\\
&= \frac{1}{T}\left(I[w_{1\ldots T}, w_{\leq 0}] - I[w_{1\ldots T}, (m_0, r_0, \ldots, r_T)]\right). && (34)
\end{aligned}$$

Now, using the calculation from (20), this can be rewritten as:

$$\begin{aligned}\mathrm{H}[w_t|m_{t-1},r_t]-\mathrm{H}[w_t|w_{<t}] =& \frac{1}{T}\left(\sum_{t=1}^{T} tI_t + T\sum_{t>T} I_t - I[w_1\dots w_T,(m_0,r_1,...,r_T)]\right)\\ =& \frac{1}{T}\left(\sum_{t=1}^{T} tI_t + T\sum_{t>T} I_t - I[w_{1\dots T},m_0] - \sum_{t=1}^{T} I[w_{1\dots T},r_t|m_0,r_{1\dots t-1}]\right).\end{aligned}$$

Due to the inequalities

$$\mathrm{I}[w_{1\dots T},m_0] \leq \mathrm{H}[m_0] \leq \sum_{t=1}^{T} tI_t \tag{35}$$

$$\mathrm{I}[w_{1\dots T},r_t|m_0,r_{1\dots t-1}] \leq \mathrm{H}[r_t] \leq \sum_{t=T+1}^{S} I_t, \tag{36}$$

this can be bounded as

$$\mathrm{H}[w_t|m_{t-1},r_t]-\mathrm{H}[w_t|w_{<t}] \geq \frac{1}{T}\left(\sum_{t=1}^{T} tI_t + T\sum_{t>T} I_t - H[m_0] - \sum_{t=1}^{T} H[r_t]\right). \tag{37}$$

Finally, this reduces as

$$\begin{aligned}\mathrm{H}[w_t|m_{t-1},r_t]-\mathrm{H}[w_t|w_{<t}] \geq& \frac{1}{T}(T\sum_{t>T} I_t - T\cdot H[r_t]) && (38)\\ =& \sum_{t>T} I_t - H[r_t] && (39)\\ \geq& \sum_{t>T} I_t - \sum_{t=T+1}^{S} I_t && (40)\\ =& \sum_{t>S} I_t. && (41)\end{aligned}$$

□

Information Locality We now show that this result predicts information locality provided that retrieving information is more expensive than keeping the same amount of information in working memory. For this, we formalize the problem of finding an optimal memory strategy as a multi-objective optimization, aiming to minimize

$$\lambda_1 H[m_t] + \lambda_2 H[r_t]. \tag{42}$$

to achieve a given surprisal level, for some setting of $\lambda_1,\lambda_2 > 0$ describing the relative cost of storage and retrieval. What is the optimal division of labor between keeping information in working memory and recovering it through retrieval? The problem

$$\min_T \lambda_1 \sum_{t=1}^{T} tI_t + \lambda_2 \sum_{t=T+1}^{S} I_t \tag{43}$$

has solution $T \approx \frac{\lambda_2}{\lambda_1}$. This means that, as long as retrievals are more expensive than keeping the same amount of information in working memory (i.e., $\lambda_2 > \lambda_1$), the optimal strategy stores information from the last $T > 1$ words in working memory. Due to the factor t inside $\sum_{t=1}^{T} tI_t$, the bound (43) will be reduced when I_t decays faster, i.e., there is strong information locality.

The assumption that retrieving information is more difficult than storing it is reasonable for cue-based retrieval models, as retrieval suffers from similarity-based interference effects due to the unstructured nature of the storage (Lewis and Vasishth, 2005). A model that maintains no information in its working memory, i.e. $H[m_t] = 0$, would correspond to a cue-based retrieval model that stores nothing in its buffers and control states, and relies entirely on retrieval to access past information. Given the nature of representations assumed in models (Lewis and Vasishth, 2005), such a model would seem to be severely restricted in its ability to parse language.

1.4 Information Locality in Language Production

Here we show results linking memory and locality in production. We show that results similar to our main theorem hold for the tradeoff between a speaker's memory and the accuracy with which they match the distribution of the language.

In the case of production, the memory–surprisal trade-off arises from the minimization of error in production of linguistic sequences. That is, given a **competence language** (a target distribution on words given contexts), a speaker tries to produce a **performance language** which is as close as possible to the competence language. The performance language operates under memory constraints, so the performance language will diverge from the competence language due to production errors. When a speaker has more incremental memory about what she has already produced, then she is able to produce linguistic sequences with less error, thus reducing the divergence between the performance language and the competence language. The reduction of this competence–performance divergence for a speaker is formally equivalent to the minimization of average surprisal for a listener.

Formally, we assign a speaker a production policy $q(w_t|m_t)$ that produces the next word conditional on the speaker's memory state m_t. We assume that speakers aim to minimize the occurrence of production errors. We formalize this as minimizing the KL divergence from the performance language $q(w_t|m_t)$ to the target competence language $p(w_t|w_{<t})$. We call this divergence the **competence–performance divergence** under the memory encoding function M and the production policy q:

$$d_M^q \equiv D_{\mathrm{KL}}[p(w_t|w_{<t})||q(w_t|m_t)] \tag{44}$$

$$= \sum_{w_{\leq t}} p(w_{\leq t}) \log \frac{p(w_t|w_{<t})}{q(w_t|m_t)}. \tag{45}$$

Under this assumption, the Information Locality Bound Theorem will apply in production as well as comprehension: The competence-performance divergence d_M^q trades off with memory load $H[m_t]$, and this tradeoff will be more favorable when languages exhibit information locality. This means that languages that exhibit information locality can be produced with greater accuracy given limited memory resources.

We derive the existence of this trade-off from the following postulates about language production. Let the competence language be represented by a stationary stochastic process, parameterized by a probability distribution $p(w_t|w_{<t})$ giving the conditional probability of any word w_t given an unbounded number of previous words. Our postulates describe a speaker who tries to find a performance language $q(w_t|m_t)$ to match the the competence language using incremental memory representations m_t:

1. Production Postulate 1 (Incremental memory). At time t, the speaker has an incremental **memory state** m_t that contains (1) her stored information about previous words that she has produced, and (2) information about her production target. The memory state is given by a **memory encoding function** M such that $m_t = M(w_{t-1}, m_{t-1})$.

2. Production Postulate 2 (Production policy). At time t, the speaker produces the next word w_t conditional on her memory state by drawing from a probability distribution $q(w_t|m_t)$. We call q the speaker's **production policy**.

3. Production Postulate 3 (Minimizing divergence). The production policy q is selected to minimize the KL divergence from the performance language to the target competence language $p(w_t|w_{<t})$. We call this divergence the **competence–performance divergence** under the memory encoding function M and the production policy q:

$$d_M^q \equiv D_{\mathrm{KL}}[p(w_t|w_{<t})||q(w_t|m_t)] \tag{46}$$

$$= \sum_{w_{\leq t}} p(w_{\leq t}) \log \frac{p(w_t|w_{<t})}{q(w_t|m_t)}. \tag{47}$$

Completing the link with the memory–surprisal trade-off in comprehension, we note that when the production policy $q(w_t|m_t)$ is selected to minimize the competence–performance divergence d_M^q, then this divergence becomes equal to the memory distortion $S_M - S_\infty$ discussed in the context of comprehension costs. Therefore, under these postulates, the Information Locality Bound Theorem will apply in production as well as comprehension (see Section 1.4.1 for formal statement and proof). This means that languages that exhibit information locality can be produced with greater accuracy given limited memory resources.

In the case of language comprehension, the trade-off represented excess processing *difficulty* arising due to memory constraints. In the case of language production, the trade-off represents *production error* arising due to memory constraints. When memory is constrained, then the speaker's productions will diverge from her target language. And as memory is more and more constrained, this divergence will increase more and more. The degree of divergence is measured in the same units as surprisal, hence the formal equivalence between the listener's and speaker's memory–surprisal trade-offs.

Although the memory–surprisal trade-off is mathematically similar between comprehension and production, it is not necessarily identical. The comprehender's memory–surprisal trade-off has to do with the amount of predictive information I_t stored in memory, where I_t is defined in terms of a probability distribution on words given t words of context. In the producer's memory–surprisal tradeoff, this probability distribution may be different, because the producer has knowledge of a production target (Production Postulate 1). Nevertheless, if the producer's probability distribution is similar to the comprehender's, then we predict the same trade-off for the producer as for the comprehender.

It may be possible to use this asymmetry to distinguish whether word and morpheme order is more optimized for the comprehender or the producer. If word order is best predicted under a probability model that uses zero information about a production target (as in the current work), then we have evidence that the comprehender's trade-off is more important. On the other hand, if word order is best predicted under a probability model that uses (partial) information about a production target, then we have evidence that the producer's trade-off is more important. As estimating the difference between these probabilility distributions is difficult, we leave this avenue of research to future work.

1.4.1 Information Locality Theorem in Production

Here, we prove an Information Locality Theorem in production. Following the Production Postulates 1–3, we consider a setting in which a speaker produces sentences with bounded memory, and analyze the deviation of the produced distribution from the actual distribution of the language. We consider a speaker who maintains memory representations and incrementally produces based on these representations:

$$P_{\text{produced}}(w_t|w_{<t}) = q(w_t|m_t) \tag{48}$$

We show a tradeoff between the memory capacity $\mathrm{H}[m_t]$ and the KL-divergence between the actual language statistics and the speaker's production distribution, as defined in Production Postulate 3:

$$d_M^q = D_{KL}(P_{\text{language}}||P_{\text{produced}}) = \mathbb{E}_{w_{<t}} \sum_{w_t} p(w_t|w_{<t}) \log \frac{p(w_t|w_{<t})}{p_{produced}(w_t|w_{<t})} \tag{49}$$

As in the case of comprehension, we model $(w_t, m_t)_{t\in\mathbb{Z}}$ as stationary; however, we do *not* assume the 'No Mindreading' condition (8).

Theorem 3. *If a speaker maintains memory*

$$\mathrm{H}[m_t] \leq \sum_{i=1}^{T} tI_t, \tag{50}$$

then

$$d_M^q = D_{KL}(P_{language}||P_{produced}) \geq \sum_{t=T+1}^{\infty} I_t. \tag{51}$$

While this bound only considers the production of a single word, it entails a bound on the production accuracy for sequences:

$$D_{KL}(P_{language}(w_1 \dots w_t|w_{\leq 0})||P_{produced}(w_1 \dots w_t|w_{\leq 0})) = t \cdot D_{KL}(P_{language}(w_1|w_{\leq 0})||P_{produced}(w_1|w_{\leq 0})) \tag{52}$$

Proof. We rewrite the KL-Divergence so that we can reduce this result to the proof in the comprehension setting (Section 1.2). First note

$$D_{KL}(P_{language}||P_{produced}) = \mathbb{E}_{w_{<t}} \left[\sum_{w_t} p(w_t|w_{<t}) \log \frac{p(w_t|w_{<t})}{p_{produced}(w_t|w_{<t})} \right] \tag{53}$$

$$= \mathbb{E}_{w_{<t}} \left[\sum_{w_t} p(w_t|w_{<t}) \log \frac{p(w_t|w_{<t})}{p(w_t|M(w_{<t}))} \right] \tag{54}$$

$$= \mathbb{E}_{w_{<t}} \left[\sum_{w_t} p(w_t|w_{<t}) \log p(w_t|w_{<t}) \right] - \mathbb{E}_{w_{<t}} \left[\sum_{w_t} p(w_t|w_{<t}) \log p(w_t|M(w_{<t})) \right] \tag{55}$$

$$= \mathrm{H}[w_t|M(w_{<t})] - \mathrm{H}[w_t|w_{<t}] \tag{56}$$

We now note that the proof in Section 1.2 can be used, without further modification, to show that

$$\mathrm{H}[w_t|M(w_{<t})] - \mathrm{H}[w_t|w_{<t}] \geq \sum_{t=T+1}^{\infty} I_t \tag{57}$$

completing the proof. The reason we can apply the proof from Section 1.2 is that Comprehension Postulate 1, where it is used in that proof, can be replaced by the analogous Production Postulate 1. □

1.5 Proof of Left-Right Invariance

Here we show that the bound provided by the Information Locality Theorem is invariant under reversal of the process. That is: Given a process $(X_t)_{t\in\mathbb{Z}}$, we define its reverse process $(Y_t)_{t\in\mathbb{Z}}$ by $Y_t := X_{-t}$. We claim that the theorem provides the same bounds for the memory-surprisal tradeoff curves. To prove this, we note:

$$I[X_t, X_0|X_{1\dots t-1}] = I[Y_{-t}, Y_0|Y_{1-t\cdots-1}] = I[Y_0, Y_t|Y_{1\dots t-1}] = I[Y_t, Y_0|Y_{1\dots t-1}] \tag{58}$$

The first step follows from the definition of Y. The second step follows from the fact that X_t, and thus also Y_t, is stationary, and thus adding t to each index in the expression does not change the resulting value. The third step uses the fact that mutual information is symmetric.

2 Examples with Analytical Calculations

Here, we provide examples of the Information Locality Theorem in settings where analytical calculations are possible. These examples are artificial and intended to demonstrate the mathematical possibility of certain phenomena; we do not intend these examples to model any linguistic phenomena.

2.1 Window-Based Model not Optimal

Here we provide an example of a stochastic process where a window-based memory encoding is not optimal, but the bound provided by our theorem still holds. This is an example where the bound provided by the theorem is not tight: while it bounds the memory-surprisal tradeoff of all possible listeners, the bound is 'optimistic', meaning that no mathematically possible memory encoding function M can exactly achieve the bound.

Let k be some positive integer. Consider a process $x_{t+1} = (v_{t+1}, w_{t+1}, y_{t+1}, z_{t+1})$ where

1. The first two components consist of fresh random bits. Formally, v_{t+1} is an independent draw from $Bernoulli(0.5)$, independent from all preceding observations $x_{\leq t}$. Second, let w_{t+1} consist of $2k$ many such independent random bits (so that $H[w_{t+1}] = 2k$)

2. The third component *deterministically* copies the first bit from $2k$ steps earlier. Formally, y_{t+1} is equal to the first component of x_{t-2k+1}

3. The fourth component *stochastically* copies the second part (consisting of $2k$ random bits) from one step earlier. Formally, each component $z_{t+1}^{(i)}$ is determined as follows: First take a sample $u_{t+1}^{(i)}$ from $Bernoulli(\frac{1}{4k})$, independent from all preceding observations. If $u_{z+1}^{(i)} = 1$, set $z_{t+1}^{(i)}$ to be equal to the second component of $w_t^{(i)}$. Otherwise, let $z_{t+1}^{(i)}$ be a fresh draw from $Bernoulli(0.5)$.

Predicting observations optimally requires taking into account observations from the $2k$ last time steps.

We show that, when approximately predicting with low memory capacities, a window-based approach does *not* in general achieve an optimal memory-surprisal tradeoff.

Consider a model that predicts x_{t+1} from only the last observation x_t, i.e., uses a window of length one. The only relevant piece of information in this past observation is w_t, which stochastically influences z_{t+1}. Storing this costs $2k$ bit of memory as w_t consists of $2k$ draws from $Bernoulli(0.5)$. How much does

it reduce the surprisal of x_{t+1}? Due to the stochastic nature of z_{t+1}, it reduces the surprisal only by about $I[x_{t+1}, w_t] = I[z_{t+1}, w_t] < 2k \cdot \frac{1}{2k} = 1$, i.e., surprisal reduction is strictly less than one bit. [1]

We show that there is an alternative model that strictly improves on this window-based model: Consider a memory encoding model that encodes each of $v_{t-2k+1}, \ldots, v_t$, which costs $2k$ bits of memory – as the window-based model did. Since $y_{t+1} = v_{t-2k+1}$, this model achieves a surprisal reduction of $H[v_{t-2k+1}] = 1$ bit, strictly more than the window-based model.

This result does not contradict our theorem because the theorem only provides *bounds* across models, which are not necessarily achieved by a given window-based model. In fact, for the process described here, no memory encoding function M can exactly achieve the theoretical bound described by the theorem.

2.2 Tight Bound for Retrieval Model

Here, we provide an example where our bound is tight for the retrieval-based model (Section 1.3) even though it is quite loose for the capacity model. That means, while no memory encoding function can exactly achieve the bound in the *capacity-bounded* setting for this particular stochastic process, there are *retrieval-based* memory encoding functions that exactly achieve the bound in the retrieval-based setting.

Defining the Process Let k be a positive integer. Consider a process $x_{t+1} = (y_{t+1}, z_{t+1}, u_{t+1}, v_{t+1})$ where

1. y_{t+1} consists of $2k$ random bits.
2. z_{t+1} is a draw from $Bernoulli(\frac{1}{4k^2})$.
3. u_{t+1} consists of $2k$ random bits if $z_t = 0$ and is equal to y_{t-2k+1} else.
4. $v_{t+1} := z_t$

Informally, z_t indicates whether u_{t+1} is copied from the past or a fresh sample; large values of k correspond to the setting where copying from the past only happens rarely.

Capacity Model We analyze the memory-surprisal tradeoff in the situation where prediction is optimal. Predicting observations $x_{t+1}, x_{t+2}, \ldots$ optimally from the past requires storing $y_{t-2k+1}, \ldots, y_t$ and z_t. This amounts to

$$H_M = (2k+1) \cdot 2k + H_2[1/4k^2] \geq 4k^2 \tag{59}$$

bits of memory in the capacity-based model, where $H_2[p] := -(p \log p + (1-p)\log(1-p))$.

We now ealuate I_t. We have

$$I_1 = I[v_{t+1}, z_t] = H_2[1/4k^2] \tag{60}$$

$$I_{2k} = I[x_{t+1}, x_{t-2k+1} | x_{t-2k+2} \ldots x_t] = I[u_{t+1}, y_{t-2k+1} | z_{t+1}] = \frac{1}{4k^2} I[u_{t+1}, y_{t-2k+1} | z_{t+1} = 1] = \frac{2k}{4k^2} = \frac{1}{2k} \tag{61}$$

and all other values of I_t are zero.

[1] We can evaluate $I[z_{t+1}, w_t]$ as follows. Set $l = k/4$. Write z, w for any of the $2k$ components of z_{t+1}, w_t, respectively. First, calculate $p(z=1|w=1) = 1/l + (1-1/l)\frac{1}{2} = 1/(2l) + 1/2 = \frac{1+l}{2l}$ and $p(z=0|w=1) = (1-1/l)\frac{1}{2} = 1/2 - 1/2l = \frac{l-1}{2l}$. Then $I[Z,W] = D_{KL}(p(z|w=1)||p(z)) = \frac{1+l}{2l}\log\frac{\frac{1+l}{2l}}{1/2} + \frac{l-1}{2l}\log\frac{\frac{l-1}{2l}}{1/2} = \frac{1+l}{2l}\log\frac{1+l}{l} + \frac{l-1}{2l}\log\frac{l-1}{l} \leq \frac{1+l}{l}\log\frac{1+l}{l} = (1+1/l)\log(1+1/l) \leq (1+1/l)(1/l) = 1/l + 1/l^2 < 2/l = \frac{1}{2k}$.

Therefore, the theorem bounds the memory cost, in the limit of perfect prediction ($T \to \infty$), only by

$$H_M \geq \sum_{t=1}^{\infty} tI_t = 2kI_{2k} = 1 \tag{62}$$

compared to a true cost $H_M \geq 4k^2$. The bound provided by the theorem is therefore loose in this case for the capacity-based model.

Retrieval Model However, it is tight for the retrieval-based model. Again, we show this in the setting of optimally precise prediction. We use

$$s_t := (y_{t-2k+1}, \ldots, y_t) \tag{63}$$

$$m_{t+1} := z_t \tag{64}$$

Then, if $z_t = 1$, we retrieve

$$r_t = g(x_{t-1}, m_{t-1}, s_{t-1}) := y_{t-2k+1} \tag{65}$$

Otherwise, if $z_t = 0$, we retrieve nothing. The cost of storing z_t is $H_2[1/4k^2]$, and the cost of retrieving r_t is $\frac{1}{4k^2} \cdot 2k = \frac{1}{2k}$.

In total, $H[m_t] = H_2[1/4k^2]$ and $H[r_t] = 1/2k$.

Taking, in the theorem, $T = 1$ and $S \to \infty$, we obtain

$$H[m_t] \geq I_1 = H_2[1/4k^2] \tag{66}$$

$$H[r_t] \geq I_{2k} = 1/2k \tag{67}$$

Thus, the bound is tight for both working memory and retrieval costs.

Furthermore, the bound provided by the theorem for the capacity-based model, while it can be loose for specific processes, is the tightest possible bound that only depends on the values of I_t. As the retrieval-based model is a generalization of the capacity-based model, it may be possible for the retrieval-based model to achieve the bound provided by the theorem even in cases when it is not possible for the capacity-based model.

2.3 Low memory requirements do not imply decay of unconditional mutual information

Our theoretical results link the memory-surprisal tradeoff to the values of *conditional* mutual information I_t, whereas prior work on the statistics of language has considered *unconditional* mutual information $I[w_t, w_0]$. Here, we show that the decay of unconditional mutual information is not necessarily linked to memory demands.

First, there are processes where unconditional mutual information does not decay with distance, even though memory load is small. Consider the constant process where with probability 1/2 all $w_t = 0$, and with probability 1/2 all $w_t = 1$. The unconditional mutual information is $I[w_t, w_0] = 1$ at all distances t, so does not decay at all. However, predicting the process optimally only requires 1 bit of memory. This is correctly captured by the Information Locality Theorem, as $I_1 = 1$ and $I_t = 0$ for $t > 1$, so $\lim_{T\to\infty} \sum_{t=1}^{T} tI_t = 1$.

Second, one can construct processes where the unconditional mutual informations $I[w_t, w_0]$ are zero for all distances t, but where optimal prediction requires nonzero memory: Consider the process consisting of 2 random bits and their XOR (called RRXOR by Crutchfield and Feldman, 2003). This one has nonzero I_2, but zero unconditional mutual information $I[w_t, w_0]$ at all distances t. Conditional mutual information is not zero, however, and – in accordance with the Information Locality Theorem – optimal prediction requires at least $\lim_{T\to\infty} \sum_{t=1}^{T} tI_t > 0$ bits of memory (Crutchfield and Feldman, 2003).

3 Study 2

3.1 Corpus Size per Language

Language	Training	Held-Out	Language	Training	Held-Out
Afrikaans	1,315	194	Indonesian	4,477	559
Amharic	974	100	Italian	17,427	1,070
Arabic	21,864	2,895	Japanese	7,164	511
Armenian	514	50	Kazakh	947	100
Bambara	926	100	Korean	27,410	3,016
Basque	5,396	1,798	Kurmanji	634	100
Breton	788	100	Latvian	4,124	989
Bulgarian	8,907	1,115	Maltese	1,123	433
Buryat	808	100	Naija	848	100
Cantonese	550	100	North Sami	2,257	865
Catalan	13,123	1,709	Norwegian	29,870	4,639
Chinese	3,997	500	Persian	4,798	599
Croatian	7,689	600	Polish	6,100	1,027
Czech	102,993	11,311	Portuguese	17,995	1,770
Danish	4,383	564	Romanian	8,664	752
Dutch	18,310	1,518	Russian	52,664	7,163
English	17,062	3,070	Serbian	2,935	465
Erzya	1,450	100	Slovak	8,483	1,060
Estonian	6,959	855	Slovenian	7,532	1,817
Faroese	1,108	100	Spanish	28,492	3,054
Finnish	27,198	3,239	Swedish	7,041	1,416
French	32,347	3,232	Thai	900	100
German	13,814	799	Turkish	3,685	975
Greek	1,662	403	Ukrainian	4,506	577
Hebrew	5,241	484	Urdu	4,043	552
Hindi	13,304	1,659	Uyghur	1,656	900
Hungarian	910	441	Vietnamese	1,400	800

Table 2: Languages, with the number of training and held-out sentences available.

3.2 Details for Neural Network Models

The network is parameterized by a vector θ of weights determining how the activations of neurons propagate through the network (Hochreiter and Schmidhuber, 1997). Given a corpus, the numeral parameters of the LSTM are chosen so as to minimize the average surprisal across the training corpus. At the beginning of training, the parameters θ are randomly initialized to some setting θ_0.

The training corpus is chopped into word sequences $w_1 \ldots w_{T_{max}}$ of length T_{max}, where T_{max} is the highest T for which we estimate I_T. We use Stochastic Gradient Descent to optimize the parameters θ so as to

minimize the surprisal

$$\frac{1}{T_{max}} \sum_{i=1}^{T_{max}} \log p_\theta(w_i|w_1 ... w_{i-1}) \tag{68}$$

When calculating the parameter update, we use three standard methods of regularization that have been shown to improve neural language modeling: dropout (Srivastava et al., 2014), word dropout, and word noising (Xie et al., 2017).

Once all sequences have been processed, we start another pass through the training data. Before each pass through the training data, the order of sentences of the training data is shuffled, and the corpus is again chopped into sequences of length T. After each pass through the training data, the average surprisal (68) at the current parameter setting θ is evaluated on the held-out partition. We terminate training once this held-out surprisal does not improve over the one computed after the previous pass any more.

In our experiments, we chose $T_{max} = 20$. Prior work has found that the probabilities $p(w_t|w_1 ... w_{t-1})$ are dominated by a small number of preceding words (Daniluk et al., 2017), suggesting that I_t will be close to zero for t greater than 20.

3.2.1 Choice of Hyperparameters

The LSTM model has a set of numerical *hyperparameters* that need to be specified before parameter estimation, such as the number of neurons and the learning rate. For each corpus, we used Bayesian optimization using the Expected Improvement acquisition function (Snoek et al., 2012) to find a good setting of the hyperparameters. We optimized the hyperparameters to minimize average surprisal (68) on the held-out partition resulting at the end of parameter estimation, on languages generated from random word order grammars. This biases the hyperparameters towards modeling counterfactual grammars better, biasing them *against* our hypothesis that real orders result in better memory-surprisal tradeoffs than counterfactual orders.

Due to reasons of computational efficiency, neural language models can only process a bounded number of distinct words in a single language (Mikolov et al., 2010). For each corpus, we limited the number of distinct processed words to the $N = 10,000$ most common words in the training corpus, a common choice for neural language models. We represented other words by their part-of-speech tags as annotated in the corpora. This applied to 37 languages, affecting an average of 11 % of words in these languages. We believe that this modeling limitation does not affect our results for the following reasons. First, this affects the same words in real and counterfactually ordered sentences. Second, all excluded words are extremely infrequent in the available data, occurring less than 10 times (except for Czech and Russian, the languages for which we have by far the largest datasets). Many of the excluded words occur only once in the dataset (78 % on average across the affected languages). This means that any model would only be able to extract very limited information about these words from the available training data, likely *less* than what is provided by the part-of-speech tag. Third, traditional N-gram models, which do not have this limitation, provide results in qualitative agreement with the neural network-based estimates.

3.2.2 Estimation of average surprisal

As described in the main paper, the mutual information I_t is estimated from entropies obtained with Markov models:

$$S_t = H[w_t|w_0, \ldots, w_{t-1}]$$

We estimate these entropies as follows. After estimating the parameter vector θ, we compute the following (T ranging from T_{max} up to the length of the held-out partition) in the held-out partition:

$$\widehat{S_T} = \frac{1}{|HeldOut| - T} \sum_{i=T}^{|HeldOut|} \log P_\theta[w_t | w_{t-T}, w_{t-T+1}, ..., w_{t-1}] \tag{69}$$

where $|HeldOut|$ is the number of words in the held-out set.

For larger values of T, the model may overfit, leading to estimates where $\widehat{S_T}$ may *increase* as the context size increases. Such a situation is an artifact of overfitting, and cannot happen for the true entropies S_t. Directly estimating I_t from $\widehat{S_T}$ would lead to negative estimates of I_t, again impossible for the true values of this quantity. We eliminate this pathological behavior by only estimating

$$S_t \approx \min_{s \leq t} \widehat{S}_s, \tag{70}$$

which amounts to only considering higher-order models $P_\theta[w_t | w_{t-T}, w_{t-T+1}, ..., w_{t-1}]$ when they improve over lower-order ones. This procedure ensures that $\hat{S}_t$ can only decrease as the context size t increases.

For each language, we collected data from the actual orderings and from several random grammars. We collect multiple samples for the actual orderings to control for variation due to the random initialization of the neural network. For each of the random grammars, we collect one sample. Data is collected according to a precision-based stopping criterion described in Section 3.2.3.

We estimate the unigram entropy $H[w_0]$ by averaging over all model runs on a given corpus.

3.2.3 Number of Samples, Precision-Based Stopping Criterion

Training neural language models is computationally costly. Therefore, we used a precision-based stopping criterion to adaptively choose a sample size for each language. Precision-based stopping criteria offer a way to adaptively choose sample size without biasing results for or against the hypothesis of interest.

We propose a stopping criterion using a global measure of the degree of optimization of the real language. For each sample x from real orderings, we look at the proportions $N_+(x)$ of samples from the baseline languages that are more optimal than x throughout the entire range where both curves are defined, and the proportion $N_-(x)$ of baseline samples that are consistently less optimal. We estimate the quotient

$$G := \frac{\mathbb{E}_{x \sim P_1}[N_+(x)]}{\mathbb{E}_{x \sim P_1}[N_+(x) + N_-(x)]} \tag{71}$$

where P_1 is the distribution over values obtained for real orderings. We use a bootstrapped confidence interval for $\mathbb{E}[G]$ for quantifying the degree of optimization. For bootstrapping, we separately resample samples from the real language and from the baseline grammars. Due to the use of bootstrapping, the confidence intervals are not exact.

For each language, we first collected 10 data points for real orderings and 10 data points for baseline orderings. We continued obtaining new data points until the CI for G had width ≤ 0.15, or there were 100 samples from P_1 and 300 samples from P_2. Up to the end, we chose the next sample to be from P_0 with probability 2/3, and P_1 otherwise.[2]

This procedure was parallelized on several machines. In the case where the stopping criterion was reached for a language while several machines were still computing samples for this language, we did not discard those samples. Consequently, more samples were collected than necessary to reach the stopping criterion; however, in a way that does not bias our results towards or against our hypothesis.

[2]Due to a scripting error, a much higher number of samples was generated for Erzya.

3.3 Samples Drawn per Language

Language	Base.	Real	Language	Base.	Real
Afrikaans	13	10	Indonesian	11	11
Amharic	137	10	Italian	10	10
Arabic	11	10	Japanese	25	15
Armenian	140	76	Kazakh	11	10
Bambara	25	29	Korean	11	10
Basque	15	10	Kurmanji	338	61
Breton	35	14	Latvian	308	178
Bulgarian	14	10	Maltese	30	24
Buryat	26	18	Naija	214	10
Cantonese	306	32	North Sami	335	194
Catalan	11	10	Norwegian	12	10
Chinese	21	10	Persian	25	12
Croatian	30	17	Polish	309	35
Czech	18	10	Portuguese	15	55
Danish	33	17	Romanian	10	10
Dutch	27	10	Russian	20	10
English	13	11	Serbian	26	11
Erzya	846	167	Slovak	303	27
Estonian	347	101	Slovenian	297	80
Faroese	27	13	Spanish	14	10
Finnish	83	16	Swedish	31	14
French	14	11	Thai	45	19
German	19	13	Turkish	13	10
Greek	16	10	Ukrainian	28	18
Hebrew	11	10	Urdu	17	10
Hindi	11	10	Uyghur	326	175
Hungarian	220	109	Vietnamese	303	12

Figure 1: Samples drawn per language according to the precision-dependent stopping criterion.

Language	Mean	Lower	Upper	Language	Mean	Lower	Upper
Afrikaans	1.0	1.0	1.0	Indonesian	1.0	1.0	1.0
Amharic	1.0	1.0	1.0	Italian	1.0	1.0	1.0
Arabic	1.0	1.0	1.0	Japanese	1.0	1.0	1.0
Armenian	0.92	0.87	0.97	Kazakh	1.0	1.0	1.0
Bambara	1.0	1.0	1.0	Korean	1.0	1.0	1.0
Basque	1.0	1.0	1.0	Kurmanji	0.93	0.88	0.98
Breton	1.0	1.0	1.0	Latvian	0.49	0.4	0.57
Bulgarian	1.0	1.0	1.0	Maltese	1.0	1.0	1.0
Buryat	1.0	1.0	1.0	Naija	1.0	0.99	1.0

Cantonese	0.96	0.86	1.0	North Sami	0.37	0.3	0.44
Catalan	1.0	1.0	1.0	Norwegian	1.0	1.0	1.0
Chinese	1.0	1.0	1.0	Persian	1.0	1.0	1.0
Croatian	1.0	1.0	1.0	Polish	0.1	0.04	0.17
Czech	1.0	1.0	1.0	Portuguese	1.0	1.0	1.0
Danish	1.0	1.0	1.0	Romanian	1.0	1.0	1.0
Dutch	1.0	1.0	1.0	Russian	1.0	1.0	1.0
English	1.0	1.0	1.0	Serbian	1.0	1.0	1.0
Erzya	0.99	0.98	1.0	Slovak	0.07	0.03	0.12
Estonian	0.8	0.72	0.86	Slovenian	0.82	0.77	0.88
Faroese	1.0	1.0	1.0	Spanish	1.0	1.0	1.0
Finnish	1.0	1.0	1.0	Swedish	1.0	1.0	1.0
French	1.0	1.0	1.0	Thai	1.0	1.0	1.0
German	1.0	0.91	1.0	Turkish	1.0	1.0	1.0
Greek	1.0	1.0	1.0	Ukrainian	1.0	1.0	1.0
Hebrew	1.0	1.0	1.0	Urdu	1.0	1.0	1.0
Hindi	1.0	1.0	1.0	Uyghur	0.65	0.57	0.73
Hungarian	0.87	0.8	0.93	Vietnamese	1.0	0.98	1.0

Figure 2: Bootstrapped estimates for the precision-dependent stopping criterion G.

3.4 N-Gram Models

Here we show that the results of Study 2 remain robust when estimating surprisal with a simple n-gram model instead of recurrent neural networks.

3.4.1 Method

We use a version of Kneser-Ney Smoothing (Kneser and Ney, 1995). For a sequence $w_1 \dots w_k$, let $N(w_{1\dots k})$ be the number of times $w_{1\dots k}$ occurs in the training set. The unigram probabilities are estimated as

$$p_1(w_t) := \frac{N(w_t) + \delta}{|Train| + |V| \cdot \delta} \tag{72}$$

where $\delta \in \mathbb{R}_+$ is a hyperparameter. Here $|Train|$ is the number of tokens in the training set, $|V|$ is the number of types occurring in train or held-out data. Higher-order probabilities $p_t(w_t|w_{0\dots t-1})$ are estimated recursively as follows. Let $\gamma > 0$ be a hyperparameter. If $N(w_{0\dots t-1}) < \gamma$, set

$$p_t(w_t|w_{0\dots t-1}) := p_{t-1}(w_t|w_{1\dots t-1}) \tag{73}$$

Otherwise, we interpolate between t-th order and lower-order estimates:

$$p_t(w_t|w_{0\dots t-1}) := \frac{\max(N(w_{0\dots t}) - \alpha, 0.0) + \alpha \cdot \#\{w : N(w_{0\dots t-1}w) > 0\} \cdot p_{t-1}(w_t|w_{1\dots t-1})}{N(w_{0\dots t-1})} \tag{74}$$

where $\alpha \in [0,1]$ is also a hyperparameter. Kneser and Ney (1995) show that this definition results in a well-defined probability distribution, i.e., $\sum_{w \in V} p_t(w|w_{0\dots t-1}) = 1$.

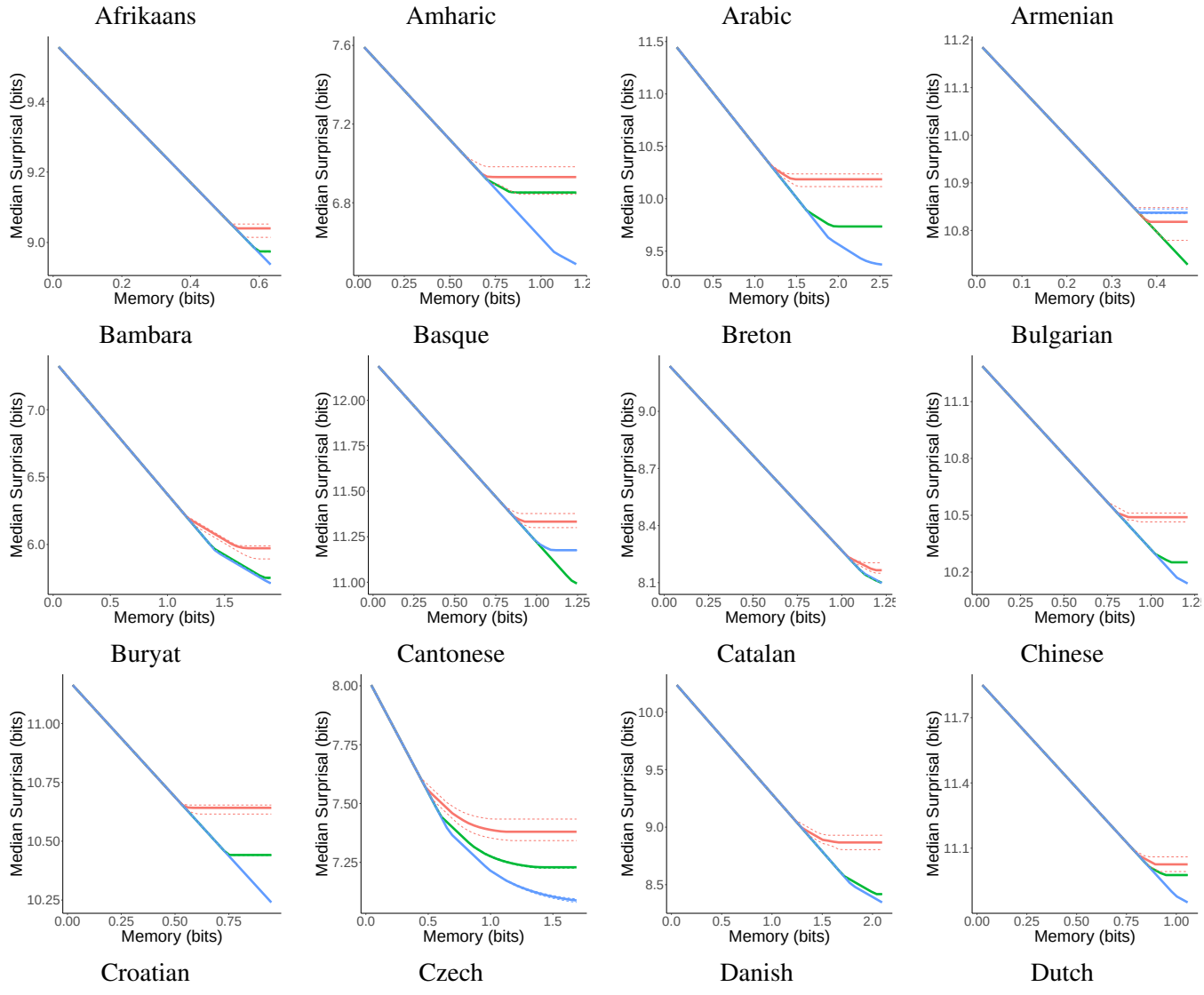
Hyperparameters α, γ, δ are tuned using the held-out set, with the same strategy as for the neural network models.

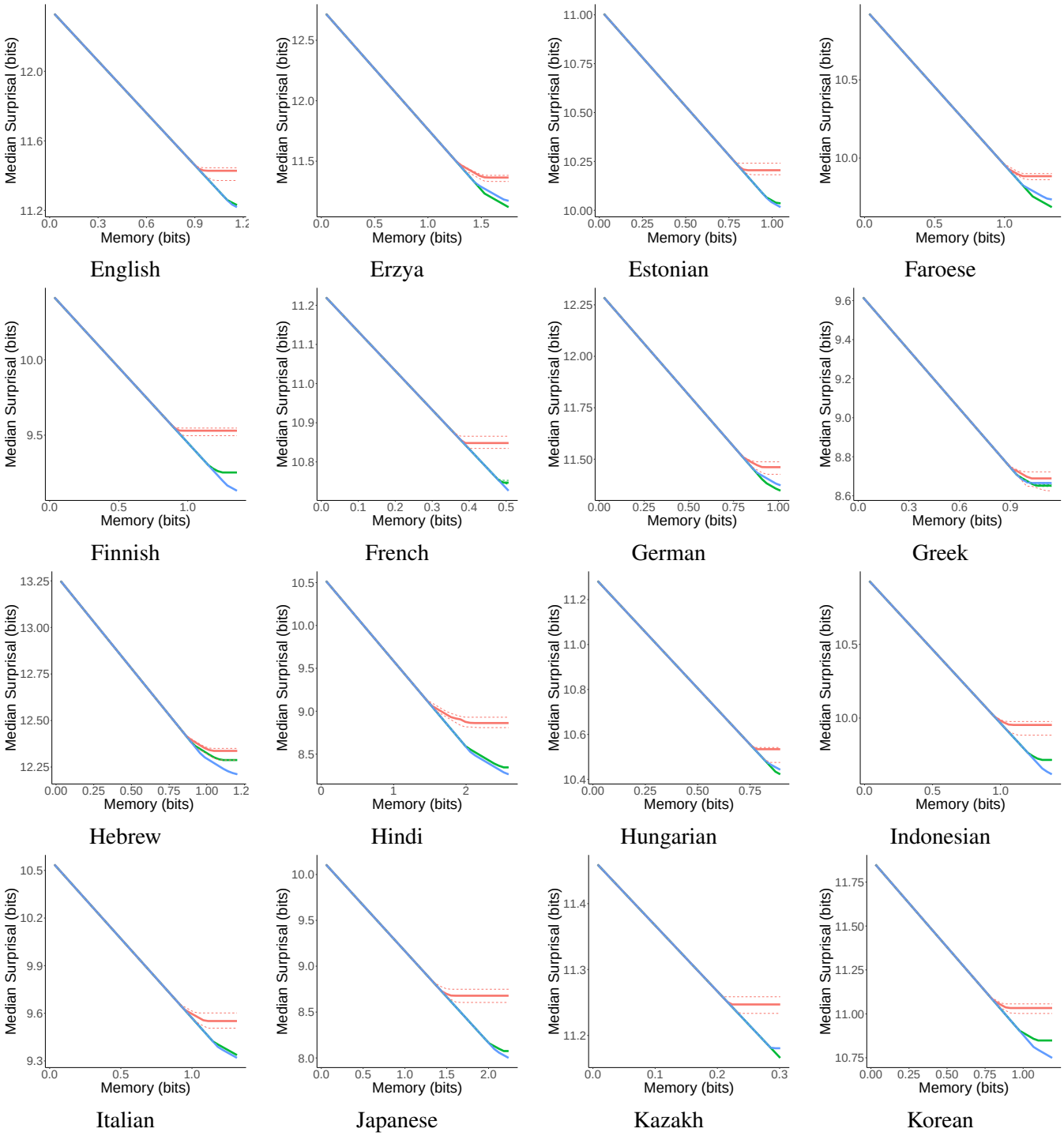
3.4.2 Results

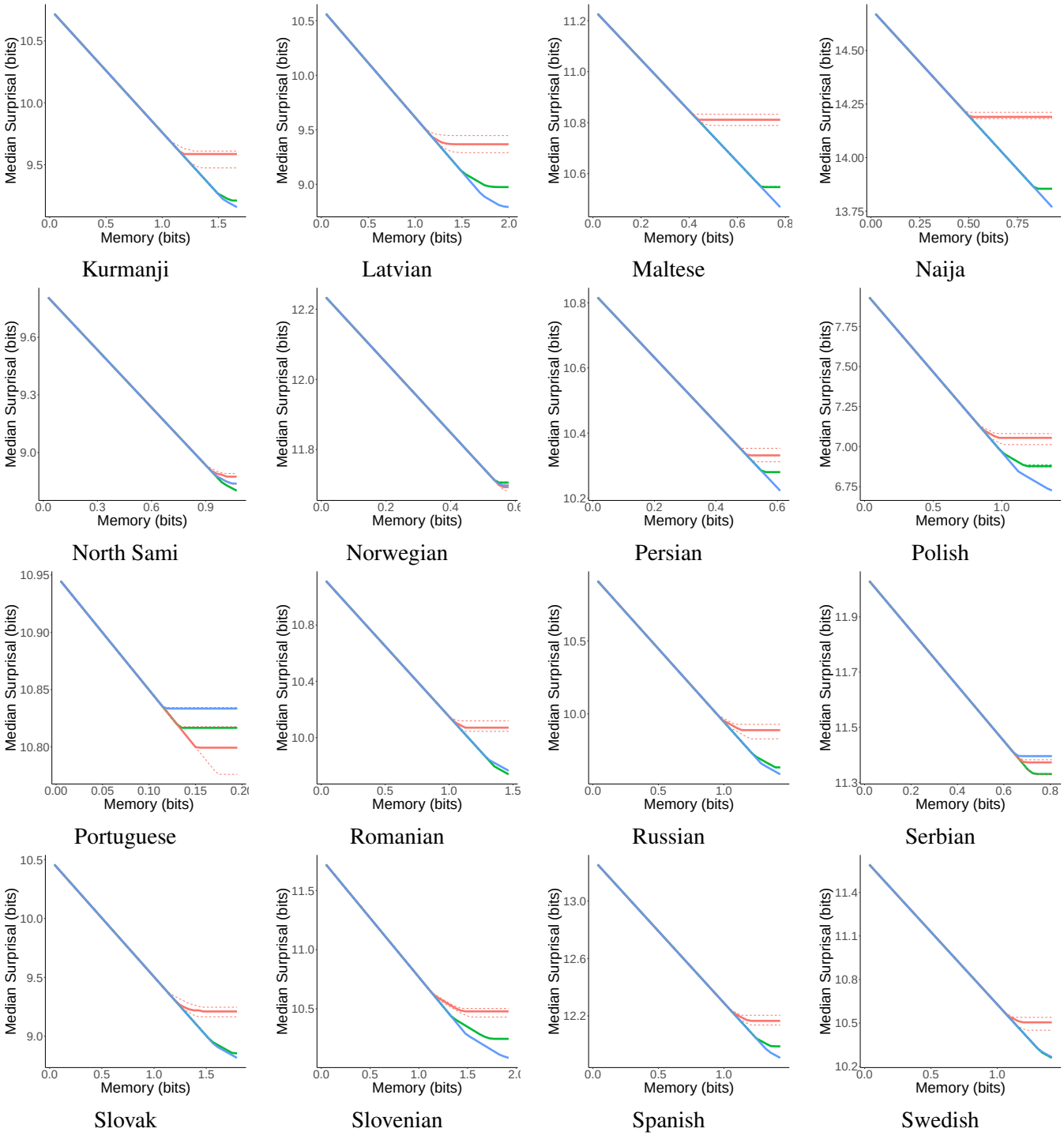
Resulting tradeoff curves are shown in Figure 3, for real orders (blue), random baselines (red), and ordering grammars fitted to the observed orders (green).

In five languages (Polish, Slovak, North Sami, Armenian, Latvian), AUC is numerically higher for the real orders than for at least 50% of baseline grammars. Among the remaining 49 languages, AUC is significantly lower than for at least 50% of baseline grammars in 46 languages at $p = 0.01$, where we controlled for multiple comparisons using Hochberg's step-up procedure. In three languages (German, Faroese, Kurmanji), the difference is numerical but not significant in this analysis. In 44 languages, the real order has lower AUC than 100% of sampled baseline grammars.

The main divergence in these results from those of the neural network-based estimator in the main paper is that a few languages with small corpora (Armenian, Faroese, Kurmanji) and a language with flexible word order (German) do not show clear evidence for optimization for the simple n-gram estimator. In the other languages, results qualitatively agree with those of the neural network-based estimator.




Median Surprisal (bits)
Memory (bits)
English
Erzya
Estonian
Faroese
Finnish
French
German
Greek
Hebrew
Hindi
Hungarian
Indonesian
Italian
Japanese
Kazakh
Korean


Median Surprisal (bits)
Memory (bits)
Kurmanji
Latvian
Maltese
Naija
North Sami
Norwegian
Persian
Polish
Portuguese
Romanian
Russian
Serbian
Slovak
Slovenian
Spanish
Swedish

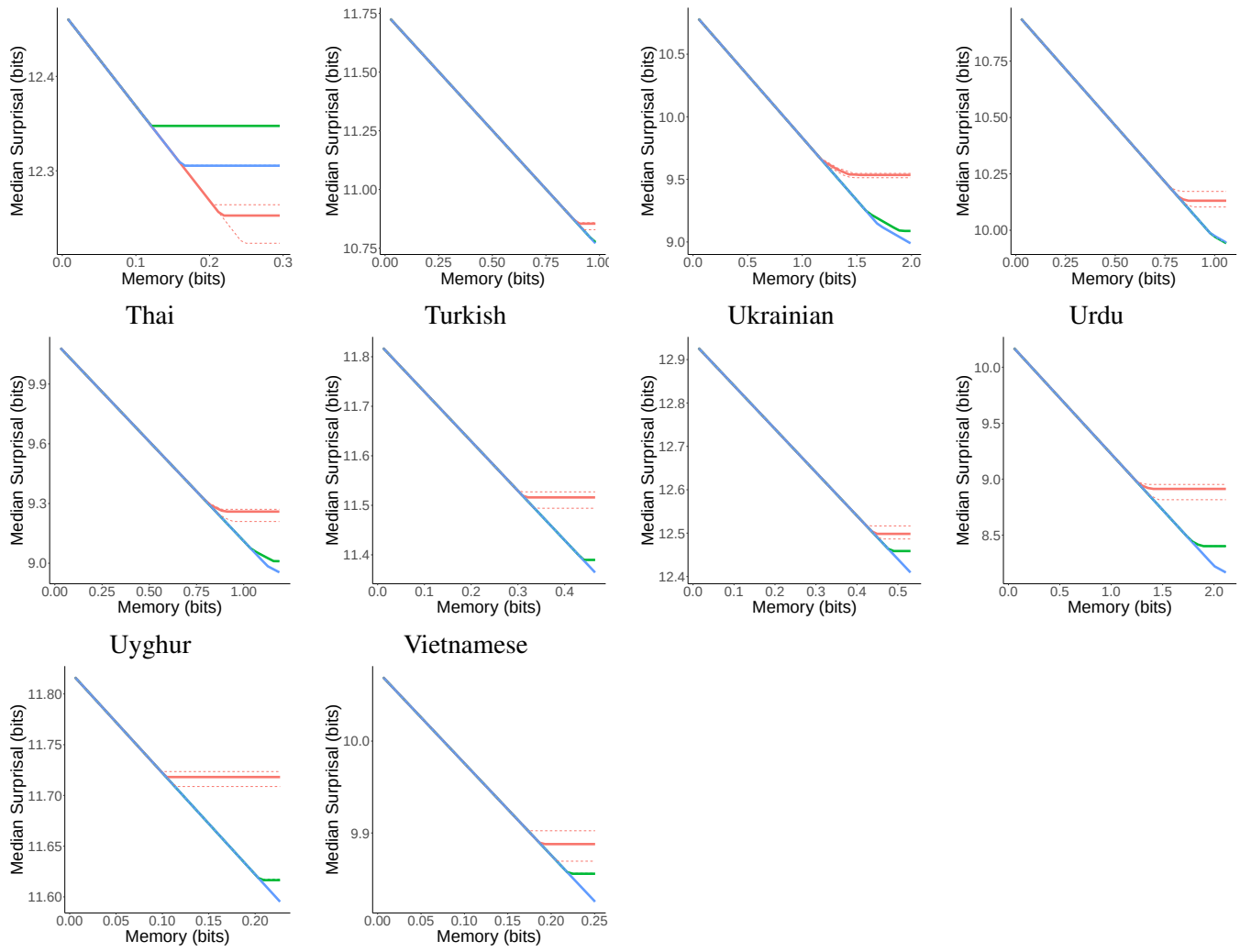


Figure 3: Memory-surprisal tradeoff curves (estimated using n-gram models): For each memory budget, we provide the median surprisal for real and random languages. Solid lines indicate sample medians for ngrams, dashed lines indicate 95 % confidence intervals for the population median. Red: Random baselines; blue: real language; green: maximum-likelihood grammars fit to real orderings.

3.5 Chart Parsing Control

LSTMs and *n*-gram models are linear sequence models that might incorporate biases towards linear order as opposed to hierarchical structure. In particular, this might bias these models towards modeling relations between elements better when they are close in linear order. Here we use chart parsing to show that the results also hold when estimating I_t using a model that is based on hierarchical structure and incorporates no bias towards linear closeness.

We use probabilistic context-free grammars (PCFG), a common formalism for representing probability distributions based on syntactic structure. PCFG surprisal is often computed in psycholinguistic research using approximate incremental parsers (Roark, 2001; Demberg et al., 2013; Schijndel et al., 2013), but

these might themselves incorporate some biases towards linear closeness due to the use of techniques such as beam-search and pruning. We instead opt for exact inference for PCFGs using chart parsing, which computes exact probabilities and surprisals for a given PCFG.

3.5.1 Deriving PCFGs from Dependency Corpora

Here, we describe how we constructed a PCFG from the training section of a dependency corpus. There is no universally accepted standard method of extracting PCFGs from dependency corpora; we chose the following procedure that tries to balance between preserving information about dependency structure and keeping the size of grammars computationally manageable.

In a first step we convert the dependency trees into binary constituent trees. We binarize so that left children branch off before right children. We assign nonterminal labels to the resulting constituents as follows. Preterminals are labeled with (1) the POS of the head, and (2) its lexical identity. We assign nonterminal labels to constituents spanning more than one word based on (1) the POS of the head, (2) the lexical identity of the head, (3) the dependency label linking head and dependent. These choices are driven by the desire to preserve information about the dependency structure in the constituent trees.

In a second step, it is necessary to reduce the number of preterminals and nonterminals, both to deal with data sparsity, and to make chart parsing tractable. In our implementation for calculating I_t (see below), we found that up to 700 nonterminals were compatible with efficient inference. (For comparison, the Berkeley parser as described by Petrov and Klein (2007) uses 1,090 nonterminals for its English grammar, while employing a highly optimized coarse-to-fine strategy that includes pruning, and thus does not provide exact inference for surprisal estimation.) We reduced the number of nonterminals as follows: (1) For words with frequency below a threshold parameter, we did not record lexical identity in preterminals and nonterminals. (2) Nonterminals that only differ in the relation label were merged if their frequency fell below a threshold parameter, (2) Nonterminals that only differ in the head's lexical identity were merged if their frequency fell below a threshold parameter. Furthermore, words occurring less than 3 times in the dataset were replaced by OOV.

An alternative method to reduce the number of nonterminals is to use merge-and-split (Petrov and Klein, 2007), but that method would have taken too long to run on all the 54 corpora.

We chose the threshold parameters for (1)-(3) separately for each language by sampling 15 configurations, and choosing the one that minimized estimated surprisal (see below) on a sampled baseline grammar, while resulting in at most 700 nonterminals and preterminals.

An alternative estimation method avoiding the binarization step would be to use the Earley parser, but that would have made it difficult to parallelize processing on GPUs (see below).

3.5.2 Estimating I_t with Chart Parsing

Calculating I_t requires estimating entropies $H[w_1, \ldots, w_t]$, and thus probabilities $P(w_1, \ldots, w_t)$. This is challenging because it requires marginalization over possible positions in a sequence. The standard parsing algorithm for binary PCFGs is the CKY algorithm; however, the standard form of this algorithm only computes the surprisal for entire sentences. There is a known extension of the CKY algorithm that calculates *prefix* probabilities (Jelinek and Lafferty, 1991; Stolcke, 1995; Goodman, 1999):

$$P[\#, X_1, \ldots, X_t] := \sum_N \sum_{Y_{1\ldots N}} P(\#, X_1, \ldots, X_t, Y_{1\ldots N}, \#) \tag{75}$$

(here, # denotes the beginning/end of a sentence), that is, the probability mass assigned to all sentences starting with the given prefix $X_1, \ldots, X_t$.

However, simultaneously summing over possible left *and* right continuations is more challenging.[3] We approach this by restricting the summation on the left to prefixes of a fixed length:

$$\sum_{Y_1 \ldots Y_N} P(\#, Y_1 \ldots Y_N, X_1, \ldots, X_t) \tag{76}$$

and estimating

$$P(X_t | X_1 \ldots X_{t-1}) \approx \mathbb{E}_{Y_1 \ldots Y_N} P(X_t | \#, Y_1 \ldots Y_N, X_1, \ldots, X_{t-1}) \tag{77}$$

Under certain conditions on the PCFG, this approximation provably converges to the true value for sufficiently large values of N. Empirically, we found that the values already became essentially stationary at $N \geq 5$.

For computational efficiency, we estimated I_t for $t = 1, \ldots 5$, finding I_t to be very close to zero for higher t. We ran the algorithm on all contiguous sequences of length $T = 5$. Following Kim et al. (2019), we took advantage of GPU parallelization for implementation of the CKY algorithm, processing 1,000 sequences in parallel.

3.5.3 Results

We computed I_t for the MLE grammar and for five random baseline grammars. We did not run this on the observed orderings, as these may have crossing branches, making binarization difficult and thus rendering comparison with baselines less meaningful.

The resulting memory-surprisal tradeoff bounds are shown in Figure 4. In most languages, a more efficient tradeoff curve is estimated for the fitted grammars than for the baseline grammars. In five languages (Finnish, Slovak, North Sami, Cantonese, Kurmanji), the fitted grammar numerically has higher AUC value than at least 50% of baseline grammars. In all other 49 languages the fitted grammar numerically has lower AUC than more than 50% of baseline grammars. In 37 languages, the fitted grammar has lower AUC than 100% of sampled baselines.

Note that absolute numbers are not comparable with other models because there are many out-of-vocabulary tokens (they are necessary because the number of non- and preterminals has to be kept low). Also, we note that the amount of exploited predictive information is much lower than in the other models, that is, the difference between surprisal at zero memory and surprisal at maximal memory is low. This agrees with the observation that PCFG independence assumptions are inadequate, and that chart parsers have not historically reached good perplexities (parsers with good perplexities such as Roark Parser and RNNGs do not make these independence assumptions, but also do not allow efficient exact chart parsing). Nonetheless, the experiment confirms the finding with a model that is based on hierarchical syntactic structure while enabling exact inference.

[3] Nederhof and Satta (2011) describe a method for calculating infix probabilities, but this method, besides being computationally costly due to construction of a large finite automaton, computes something subtly different from the quantity required here: It computes the probability mass of sentences containing a given string, not accounting for multipe occurrences of the same string in a longer sentence.

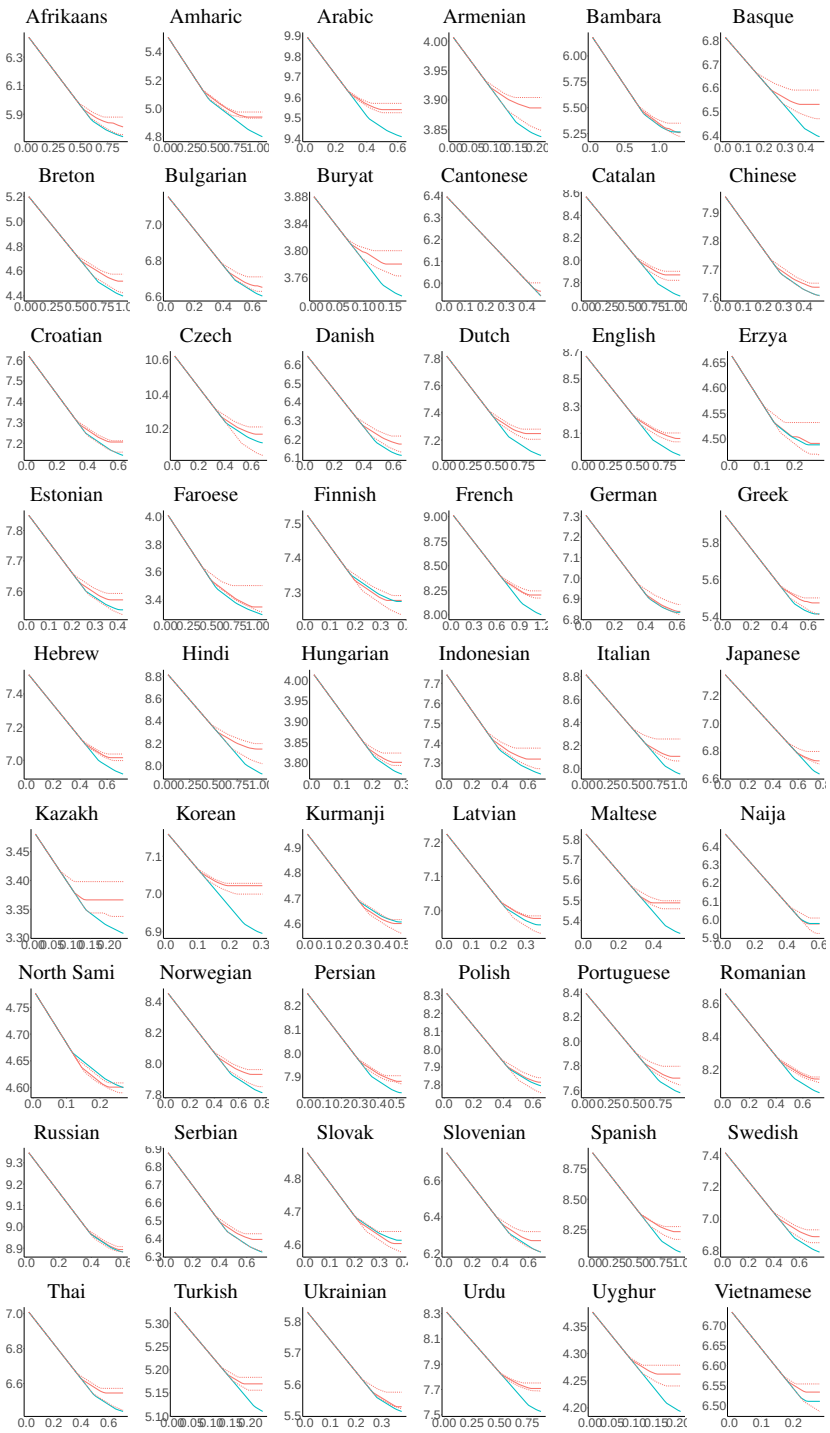


Figure 4: Memory-surprisal tradeoffs computed with the PCFG estimator, comparing fitted grammars (blue) with baselines (red). For the random baselines, we provide the sample median and 95% confidence intervals obtained with the binomial test.

3.6 Dependence on Corpus Size

Here, we examine the effect of corpus size on the estimated memory-susprisal tradeoff curves. For four languages with particularly large available datasets (Czech, English, Russian, Spanish), we repeated the estimation of the memory–surprisal tradeoff curve using 500 and 2,000 randomly selected sentences from their training sets, and using the same heldout sets as in the main experiment. These constructed datasets are smaller than available even for most languages in the main experiment: every dataset used in the main experiment has more than 500 sentences, and many languages have more than 2000 sentences available. The resulting estimates are shown in Figure 5. In each language, the absolute values of surprisal achievable at a given level of memory decrease as data increases, and the maximum level of memory at which surprisal can still be reduced further increases. Despite these differences, the relative order of the three types of orderings (fitted, real, baselines) is mostly the same across different data set sizes. For instance, in English, real orders have the most efficient curves, and baselines have the least efficient ones, across data set sizes. The only exception is the position of real orders in Czech, which are estimated to be less efficient at small training data.

4 Study 3

4.1 Determining Japanese Verb Suffixes

Here, we describe how we determined the Japanese verb suffixes described in the main paper. We determined a set of frequent morphemes as follows. We selected all morphemes occurring in the dataset at least 50 times and annotated their meaning/function. Among these, three morphemes are treated as independent words, not suffixes, by Kaiser et al. (2013) (*dekiru* 'be able to', *naru* 'become', *yoo* 'as if'); we excluded these. Furthermore, passive and potential markers are formally identical for many verbs; we included both here.

We list the morphemes according to the order extracted according to the model. Note that there is no universally accepted segmentation for Japanese suffixes; we follow the UD tokenization in choosing which suffixes to segment.[4]

1. Derivation: *-su-* (allomorphs *-suru-*, *-shi-*), derives verbs from Sino-Japanese words. This is lemmatized as *suru*.

2. VALENCE: causative (*-(s)ase-*) (Hasegawa (2014, 142), Kaiser et al. (2013, Chapter 13)). In the UD data, this is lemmatized as *saseru*, *seru* (190 occurrences).

3. VOICE: passive (*-are-*, *-rare-*) (Hasegawa (2014, 152), Kaiser et al. (2013, Chapter 12)). In the UD data, this is lemmatized as *rareru*, *reru* ($\approx$ 2000 occurrences).

4. MOOD, MODALITY:

    (a) potential (allomorphs *-are-*, *-rare-*, *-e-*). In the UD data, this is lemmatized as *rareru*, *reru*, *eru*, *keru*. This is formally identical to the passive morpheme for many verbs (Vaccari and Vaccari (1938, 346), Kaiser et al. (2013, 398)).

[4]The biggest difference to some other treatments is that the ending *-u*/*-ru* is viewed as part of the preceding morpheme that appears in some environments due to allomorphic variation, while it is viewed as a nonpast suffix in some other treatments (Hasegawa, 2014, p.116); if it were treated as a nonpast suffix, it would occupy a slot together with the past, future/hortative, and nonfiniteness affixes.

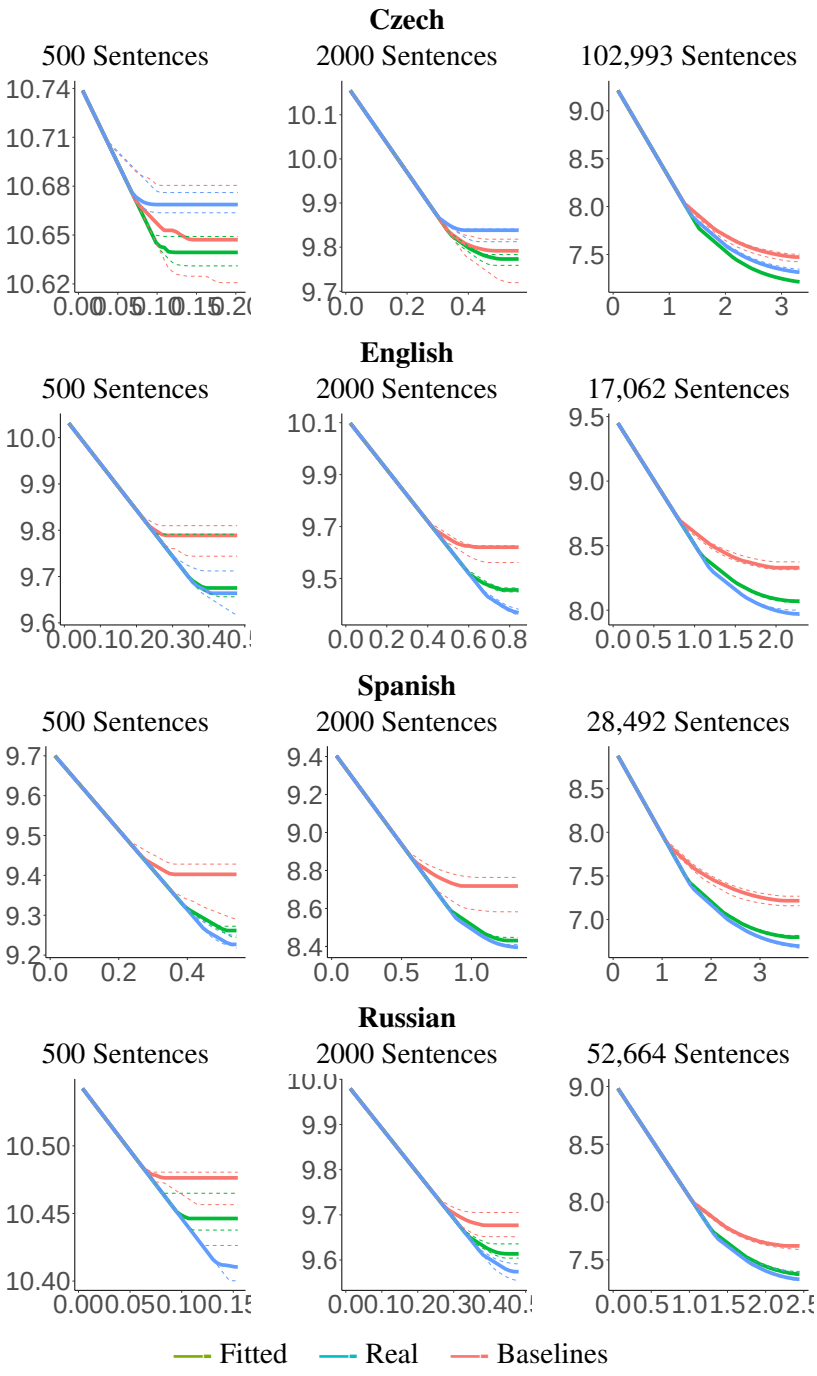


Figure 5: Dependence of estimated tradeoff curves on corpus sizes: For four languages with particularly large available datasets, we show memory–surprisal tradeoff curves estimated from 500 training sentences (left), 2000 training sentences (middle), and the full corpus (right). The x-axes show memory (in bits), the y-axes show surprisal (in bits).

(b) politeness *-mas-* (allomorphs *-masu-*, *-mashi-*, *-mase-*) (Kaiser et al., 2013, 190). In the UD data, this is lemmatized as *masu* ($\approx$ 600 occurrences).

(c) MODALITY: desiderative *-ta-* (allomorphs: *-tai*, *-taku-*, *-taka-*) (85 occurrences) (Kaiser et al., 2013, 238).

5. NEGATION: negation *-na-* (allomorphs: *-nai*, *-n-*, *-nakat-*). Lemmatized as *nai* (630 occurrences).

6. TENSE/ASPECT/MOOD:

   (a) *-ta* for past (4K occurrences) (Kaiser et al., 2013, 211)

   (b) *-yoo* for hortative, future, and similar meanings (Kaiser et al., 2013, 229). This is lemmatized as *u* (92 occurrences).

7. *-te* derives a nonfinite form (Kaiser et al., 2013, 186). (4K occurrences)

We provide examples illustrating the relative ordering of different morphemes. Note that passive and potential markers do not co-occur; we merge them here because they are not formally distinct for many verbs. We omit examples with -te; it always follows other suffixes that are compatible with it.

Stem	Caus.	Pass./Pot.	Polite.	Desid.	Neg.	TAM	
mi					naka	tta	did not see (Vaccari and Vaccari, 1938, 153)
mi				taku	nai		do not wish to see (Vaccari and Vaccari, 1938, 98)
mi				taku	naka	tta	did not wish to see (Vaccari and Vaccari, 1938, 98)
tat	ase	rare				ta	was made to stand up (Kaiser et al., 2013, 396)
waraw		are				ta	was laughed at (Kaiser et al., 2013, 384)
mi		rare	mase		n		is not seen (Vaccari and Vaccari, 1938, 337)
mi		rare	mash			yoo	will be seen (Vaccari and Vaccari, 1938, 337)
de					naka	roo	will not go out (Vaccari and Vaccari, 1938, 170)
mi		e	mase		n		cannot see (Vaccari and Vaccari, 1938, 349)

4.2 Determining Sesotho Verb Affixes

Here, we describe how we determined the Sesotho verb prefixes and suffixes. Sesotho has composite forms consisting of an inflected auxiliary followed by an inflected verb. Both verbs carry subject agreement. While they are annotated as a unit in the Demuth corpus, they are treated as separate words in grammars (Doke and Mofokeng, 1967; Guma, 1971). We separated these, taking the main verb to start at its subject agreement prefix. We only considered main verbs for the experiments here. Forms in child utterances are annotated with well-formed adult forms; we took these here. In the Demuth corpus, each morpheme is annotated; a one- or two-letter key indicates the type of morpheme (e.g. subject agreement, TAM marker). We classified morphemes by this annotation.

According to Demuth (1992), affixes in the Sesotho verb have the following order:

1. Subject agreement

2. Tense/aspect

3. Object agreement

4. Verb stem
5. 'Extension'/perfect/passive markers, where 'extension' refers to causative, neuter/stative, reversive, etc.
6. Mood

We refined this description by considering all morpheme types occurring at least 50 times in the corpus.

As in Japanese, morphemes show different forms depending on their environment. The corpus contains some instances of fused neighboring morphemes that were not segmented further; we segmented these into their underlying morphemes for modeling prediction on the level of morphemes.

Prefixes

1. Subject agreement:

   This morpheme encodes agreement with the subject, for person, number, and noun class (the latter only in the 3rd person) (Doke and Mofokeng, 1967, §395) (Guma, 1971, p. 162).

   In the Demuth corpus, this is annotated as *sm* (17K occurrences) for ordinary forms, and *sr* (193 occurrences) for forms used in relative clauses.

2. Negation:

   In various TAM forms, negation is encoded with a morpheme *-sa-* in this position (362 occurrences) (Guma, 1971, p. 172) (Doke and Mofokeng, 1967, §429). Common allomorphs in the corpus include *ska*, *seka*, *sa*, *skaba*.

3. Tense/Aspect/Mood, annotated as tˆ (13K occurrences) (Guma, 1971, p. 165)

   Common TAM markers in this position in the corpus include, with the labels provided in the Demuth corpus:

   - *-tla-*, *-tlo-*, *-ilo-* future (Doke and Mofokeng, 1967, §410–412)
   - *-a-* present (Doke and Mofokeng, 1967, §400)
   - *-ka-* potential (Doke and Mofokeng, 1967, §422–428)
   - *-sa-* persistive (Doke and Mofokeng, 1967, §413–418)
   - *-tswa-* recent past (Doke and Mofokeng, 1967, §404–406)

   In the corpus, TAM prefixes are often fused with the subsequent object marker.

4. OBJECT agreement (labeled om, 6K occurrences) or reflexive (labeled rf, 751 occurrences).

   Similar to subject agreement, object agreement denotes person, number, and noun class features of the object. Unlike subject agreement, it is optional (Doke and Mofokeng, 1967, §459).

   Object agreement and reflexive marking are mutually exclusive (Guma, 1971, p. 165).

Verb Suffixes in Sesotho Again, we extracted morpheme types occurring at least 50 times.

1. Reversive: (labeled *rv*, 214 occurrences), (Doke and Mofokeng, 1967, §345).

   This suffix changes semantics. Examples: tlama 'bind' – tlamōlla 'loosen', etsa 'do' – etsōlla 'undo' (Doke and Mofokeng, 1967, §346). Such suffixes are found across Bantu languages (Schadeberg, 2003).

2. VALENCE:

   (a) causative (labeled *c*, 1K occurrences), *-isa* (with morphophonological changes) (Doke and Mofokeng, 1967, §325)

   (b) neuter (labeled *nt*, 229 occurrences), *-eha*, *-ahala* (Doke and Mofokeng, 1967, §307)
   The neuter suffix reduces valence: lahla 'throw away' – lahlela 'get lost', sēnya 'to damage' – sēnyeha 'to get damaged' (Doke and Mofokeng, 1967, §308).

   (c) applicative (labeled *ap*, 2K occurrences) -el- (Doke and Mofokeng, 1967, §310)
   The applicative suffix increases valence: bōlela 'to say' bōlella 'to say to (s.o.)' (Doke and Mofokeng, 1967, §310).

   (d) Perfective/Completive *-ella* (annotated cl, 66 occurrences) (Doke and Mofokeng, 1967, §336)
   This does not actually change valence, but it is formally a reduplication of the applicative suffix (Doke and Mofokeng, 1967, §336), and as such its ordering behavior patterns with that of valence suffixes, in particular, it is placed before the passive suffix.[5]

   (e) Reciprocal *-ana* (annotated rc, 103 times) (Doke and Mofokeng, 1967, §338)
   This reduces valence: rata 'to love' – ratana 'to love another' (Doke and Mofokeng, 1967, §338).

   Some of these suffixes can be stacked, e.g., see (Doke and Mofokeng, 1967, §345) for reversive+causative, and (Doke and Mofokeng, 1967, §314-315) for applicative suffixes applied to other valence affixes.[6]

   Some other suffixes documented in the literature do not occur frequently or are not annotated in the corpus (e.g., the associative suffix (Doke and Mofokeng, 1967, §343)).

3. VOICE: passive *-w-* (labeled *p*, 1K occurrences) (Doke and Mofokeng, 1967, §300)

4. TENSE: tense (labeled tˆ, 3K occurrences) .

   The only tense suffix is the perfect affix *-il-*, which has a range of allomorphs depending on the preceding stem and valence/voice suffixes, if present (Doke and Mofokeng, 1967, §369), (Guma, 1971, p. 167). Common morphs in the Demuth corpus are *-il-* and *-its-*.

5. MOOD: Mood (labeled mˆ, 37K occurrences)

   In the Demuth corpus, the following mood endings are labeled (the analysis provided by Demuth (1992) is different from that provided by Doke and Mofokeng (1967), meaning the citations are only approximate):

[5]Example from the Demuth corpus: u-neh-el-ets-w-a-ng tˆp.om2s-give-ap-cl-p-mˆin-wh 'What is it that you want passed to you?'.

[6]Example of reciprocal+applicative from Demuth corpus: ba-arol-el-an-a sm2-tˆp_divide-ap-rc-mˆin 'Do they share?'

(a) Imperative (labeled IMP) (Doke and Mofokeng, 1967, §386–387): singular (-e, labeled IMP) (Doke and Mofokeng, 1967, §386) and plural (-ang, labeled IMP.PL) (Doke and Mofokeng, 1967, §386).

Similar subjunctive SBJV1 *-e* (singular), *-eng* (plural).

(b) IND (-a, -e) and NEG (-e, -a) (Doke and Mofokeng, 1967, §394–421).

(c) subjunctive SBJV2 (-e, -a) (Doke and Mofokeng, 1967, §444–455)

6. Interrogative (labeled *wh*, 2K times) and relative (labeled *rl*, 857 times) markers *-ng*.

The interrogative marker *-ng* is a clitic form of *eng* 'what' according to (Guma, 1971, p. 168), (Doke and Mofokeng, 1967, §160, 320, 714); it is treated as a suffix in the Demuth corpus.

The relative marker *-ng* is affixed to verbs in relative clauses are marked with -ng (Doke and Mofokeng, 1967, §271, 793).

Examples from Demuth (1992):

Sbj.	Obj.	V	Val.	Voice	T.	M.	
o		pheh			il	e	(Thabo) cooked (food) (Demuth (1992) (15))
ke	e	f		uw		e	(I) was given (the book) (Demuth (1992) (26c))
o		pheh	el			a	(Thabo) cooks (food for Mpho) (Demuth (1992) (41))
o		pheh	el	w		a	(Mpho) is being cooked (food) (Demuth (1992) (42))

4.3 Experiment

Identifying underlying morphemes in Japanese In Japanese, we labeled suffixes for underlying morphemes with the aid of provided lemmatization. In most cases, underlying morphemes correspond to lemmas in the UD treebank. For the causative suffix, the treebank uses the lemmas *saseru* and *seru* depending on the verb stem. As passive and potential suffixes are formally identical for many verbs, they are not fully distinguished in the treebank annotation; we collapsed them into a single underlying morpheme labeled *Passive/Potential*. It corresponds to the lemmas *reru*, *rareru*, *eru*, *keru* in the treebank annotation.

Quantifying Prediction on the Phoneme Level In the main paper, we quantified prediction on the level of morphemes. We also repeated the experiments with prediction quantified on the level of phonemes.

For Japanese, we transliterated verb forms into syllabic Hiragana with the tagger Kytea (Neubig and Mori, 2010; Neubig et al., 2011), and then automatically phonemized these syllabic representations.

For Sesotho, we use the phonological transcription provided in the Demuth corpus. The Sesotho corpus has some cases of merged forms, where neighboring morphemes are merged and not segmented further. While we represented these as the corresponding sequence of underlying morphemes when modeling morpheme prediction, we ordered these merged phonemes according to the position that a grammar assigns to its first morpheme for modeling prediction on the phoneme level.

Estimating Predictability on Training Set In the main paper, we used the heldout set to estimate the memory-surprisal tradeoff when optimizing orders for AUC. We also repeated experiments using instead the training set. In this case, we did not apply smoothing; instead, we directly computed I_t for the empirical distribution given by the training corpus. We refer to this estimation method as the 'naive' estimator, because it directly applies the definition of I_t to the distribution defined by the n-gram counts in the training set.

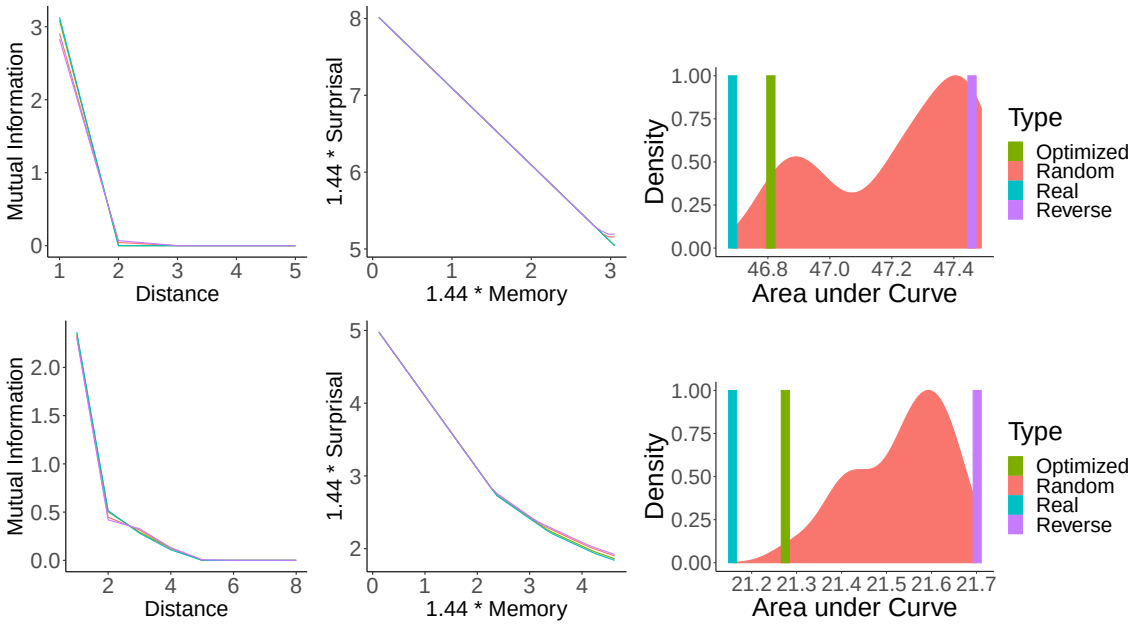


Figure 6: Japanese verb suffixes, measuring prediction on the level of morphemes (top) and phonemes (bottom), for real, random, approximately optimized, and reverse orderings. Left: I_t as a function of t. Center: Memory-surprisal tradeoff. Right: Areas under the curve for the memory-surprisal tradeoff.

Results Results for the memory-surprisal tradeoffs are shown in Figures (6–8). Accuracies on predicting orderings are shown in Figures (7-9). In the main paper, we report accuracies computed over all forms occurring in the corpus, counting each form by the number of times it occurs. This corresponds to the 'Tokens' results in Figures (7-9). Additionally, we also provide accuracies computed when counting each form only once, no matter how often it occurs; these are the 'Types' results. This method downweights high-frequency forms and upweights low-frequency forms. Results largely agree between the two methods, showing that results are not driven specifically by high-frequency forms. In Figures (7-9), we provide results both for optimizing on the heldout set as in the main paper, and for optimizing for the training set ('Naive'). Results largely agree between the two methods.

References

Bentz, C., Alikaniotis, D., Cysouw, M., and Ferrer-i Cancho, R. (2017). The entropy of words—learnability and expressivity across more than 1000 languages. *Entropy*, 19(6):275.

Cover, T. M. and Thomas, J. (2006). *Elements of Information Theory*. John Wiley & Sons, Hoboken, NJ.

Crutchfield, J. P. and Feldman, D. P. (2003). Regularities unseen, randomness observed: Levels of entropy convergence. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 13(1):25–54.

Daniluk, M., Rocktäschel, T., Welbl, J., and Riedel, S. (2017). Frustratingly short attention spans in neural language modeling. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

			Pairs	Full
Tokens	Naive	Optimized for Phoneme Prediction	0.982 (SD 0.001)	0.979 (SD 0.001)
		Optimized for Morpheme Prediction	0.93 (SD 0.011)	0.919 (SD 0.009)
	Heldout	Optimized for Phoneme Prediction	0.963 (SD 0.006)	0.958 (SD 0.006)
		Optimized for Morpheme Prediction	0.953 (SD 0.011)	0.943 (SD 0.014)
		Random Baseline	0.496 (SD 0.269)	0.415 (SD 0.271)
Types	Naive	Optimized for Phoneme Prediction	0.974 (SD 0.002)	0.969 (SD 0.002)
		Optimized for Morpheme Prediction	0.903 (SD 0.015)	0.883 (SD 0.013)
	Heldout	Optimized for Phoneme Prediction	0.948 (SD 0.009)	0.938 (SD 0.009)
		Optimized for Morpheme Prediction	0.937 (SD 0.014)	0.921 (SD 0.017)
		Random Baseline	0.496 (SD 0.269)	0.415 (SD 0.271)

Figure 7: Accuracy of approximately optimized orderings, and of random baseline orderings, in predicting verb suffix order in Japanese. 'Pairs' denotes the rate of pairs of morphemes that are ordered correctly, and 'Full' denotes the rate of verb forms where order is predicted entirely correctly. We show means and standard deviations over different runs of the optimization algorithm ('Optimized'), and over different random orderings ('Random'). 'Tokens' results are obtained by counting each form by the number of occurrences in the data set; 'Types' results count each form only once. 'Naive' models are optimized for in-sample AUC, 'Heldout' models are optimized for heldout AUC. The figures in the main paper correspond to the *Heldout + Optimized for Morpheme Prediction figures*.

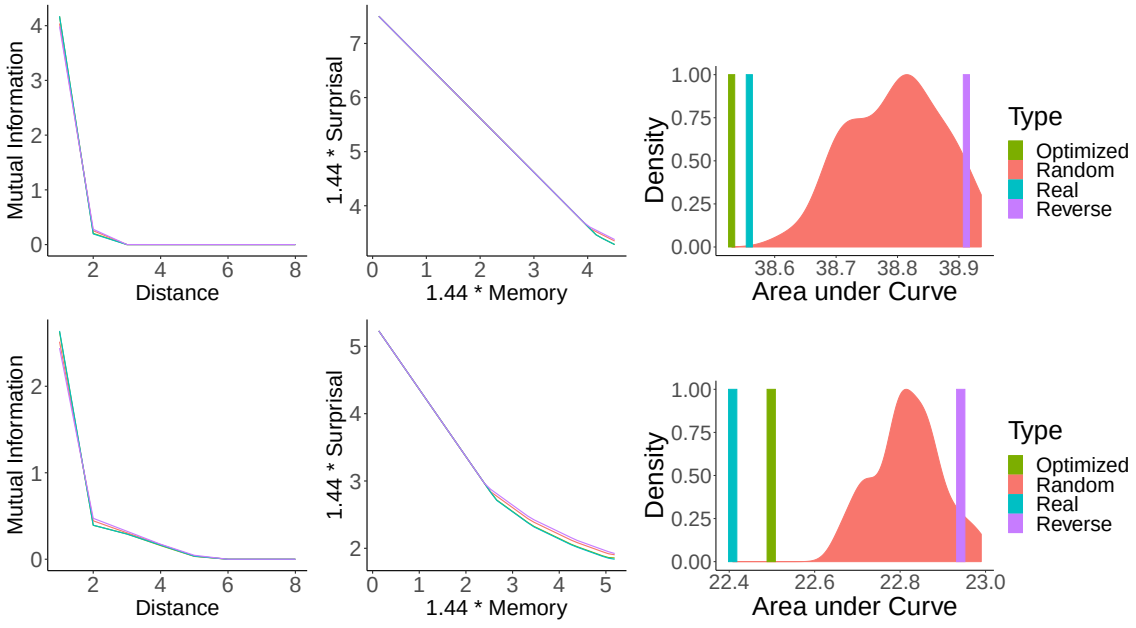


Figure 8: Sesotho verb affixes, measuring prediction on the level of morphemes (top) and phonemes (bottom), for real, random, approximately optimized, and reverse orderings. Left: I_t as a function of t. Center: Memory-surprisal tradeoff. Right: Areas under the curve for the memory-surprisal tradeoff.

				Prefixes		Suffixes	
				Pairs	Full	Pairs	Full
Tok.	Naive	Phon.	Opt.	0.985 (SD 0.0)	0.979 (SD 0.0)	0.989 (SD 0.0)	0.987 (SD 0.0)
			Rand.	0.361 (SD 0.307)	0.273 (SD 0.319)	0.431 (SD 0.198)	0.39 (SD 0.204)
		Morph.	Opt.	0.999 (SD 0.0)	0.998 (SD 0.0)	0.806 (SD 0.0)	0.723 (SD 0.0)
			Rand.	0.398 (SD 0.313)	0.303 (SD 0.319)	0.569 (SD 0.208)	0.511 (SD 0.228)
	Heldout	Phon.	Opt.	0.993 (SD 0.0)	0.989 (SD 0.0)	0.855 (SD 0.139)	0.836 (SD 0.152)
			Rand.	0.361 (SD 0.307)	0.273 (SD 0.319)	0.431 (SD 0.198)	0.39 (SD 0.204)
		Morph.	Opt.	0.99 (SD 0.0)	0.992 (SD 0.0)	0.756 (SD 0.012)	0.675 (SD 0.014)
			Rand.	0.398 (SD 0.313)	0.303 (SD 0.319)	0.569 (SD 0.208)	0.511 (SD 0.228)
Typ.	Naive	Phon.	Opt.	0.976 (SD 0.0)	0.966 (SD 0.0)	0.985 (SD 0.0)	0.98 (SD 0.0)
			Rand.	0.365 (SD 0.294)	0.267 (SD 0.296)	0.447 (SD 0.22)	0.398 (SD 0.235)
		Morph.	Opt.	0.997 (SD 0.0)	0.996 (SD 0.0)	0.844 (SD 0.0)	0.758 (SD 0.0)
			Rand.	0.405 (SD 0.308)	0.303 (SD 0.305)	0.546 (SD 0.197)	0.464 (SD 0.22)
	Heldout	Phon.	Opt.	0.988 (SD 0.0)	0.982 (SD 0.0)	0.871 (SD 0.118)	0.852 (SD 0.125)
			Rand.	0.365 (SD 0.294)	0.267 (SD 0.296)	0.447 (SD 0.22)	0.398 (SD 0.235)
		Morph.	Opt.	0.983 (SD 0.0)	0.986 (SD 0.0)	0.782 (SD 0.018)	0.697 (SD 0.02)
			Rand.	0.405 (SD 0.308)	0.303 (SD 0.305)	0.546 (SD 0.197)	0.464 (SD 0.22)

Figure 9: Accuracy of approximately optimized orderings, and of random baseline orderings, in predicting verb affix order in Sesotho. 'Pairs' denotes the rate of pairs of morphemes that are ordered correctly, and 'Full' denotes the rate of verb forms where order is predicted entirely correctly. We show means and standard deviations over different runs of the optimization algorithm ('Opt.'), and over different random orderings ('Random'). 'Tokens' resultsare obtained by counting each form by the number of occurrences in the data set; 'Types' results count each form only once. 'Naive' models are optimized for in-sample AUC on the training set, 'Heldout' models are optimized for heldout AUC.

	Real	Optimized
	Stem	Stem
1	suru	future
2	causative	desiderative
3	passive/potential	causative
4	desiderative	suru
5	politeness	passive/potential
6	negation	politeness
7	future	negation
	past	nonfinite
	nonfinite	past

Figure 10: Comparing order of Japanese affixes in the observed orders (left) and according to an approximatively optimized grammar (right), optimized for AUC on the *training* set.

	Real	Optimized
1	Subject (relative)	Subject
	Subject	Subject (relative)
2	Negation	Negation
3	Tense/aspect	Tense/aspect
4	Object	Object
	Stem	Stem
1	Reversive	Reversive
2	Causative	Reciprocal
	Neuter	Causative
	Applicative	Neuter
	Reciprocal	Applicative
3	Passive	Passive
4	Tense/aspect	Tense/aspect
5	Mood	Interrogative
6	Interrogative	Relative
	Relative	Mood

Figure 11: Comparing order of Sesotho affixes in the observed orders (left) and according to an approximatively optimized grammar (right), optimized for AUC on the *training* set. Note that order was separately optimized for prefixes and suffixes.

Debowski, L. (2011). Excess entropy in natural language: Present state and perspectives. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 21(3):037105.

Demberg, V., Keller, F., and Koller, A. (2013). Incremental, predictive parsing with psycholinguistically motivated tree-adjoining grammar. *Comput. Linguistics*, 39(4):1025–1066.

Demuth, K. (1992). Acquisition of Sesotho. In *The cross-linguistic study of language acquisition*, pages 557–638. Lawrence Erlbaum Associates.

Doke, C. M. and Mofokeng, S. M. (1967). *Textbook of southern Sotho grammar*. Longmans.

Doob, J. L. (1953). *Stochastic processes*. New York Wiley.

Ebeling, W. and Pöschel, T. (1994). Entropy and Long-Range Correlations in Literary English. *Europhysics Letters (EPL)*, 26(4):241–246.

Goodman, J. (1999). Semiring parsing. *Comput. Linguistics*, 25(4):573–605.

Guma, S. M. (1971). *An outline structure of Southern Sotho*. Shuter and Shooter.

Hahn, M. and Futrell, R. (2019). Estimating predictive rate-distortion curves via neural variational inference. *Entropy*, 21(7):640.

Hasegawa, Y. (2014). *Japanese: A linguistic introduction*. Cambridge University Press.

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

Jelinek, F. and Lafferty, J. D. (1991). Computation of the probability of initial substring generation by stochastic context-free grammars. *Comput. Linguistics*, 17(3):315–323.

Kaiser, S., Ichikawa, Y., Kobayashi, N., and Yamamoto, H. (2013). *Japanese: A comprehensive grammar*. Routledge.

Kim, Y., Dyer, C., and Rush, A. M. (2019). Compound probabilistic context-free grammars for grammar induction. In Korhonen, A., Traum, D. R., and Màrquez, L., editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2369–2385. Association for Computational Linguistics.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE.

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

Lin, H. W. and Tegmark, M. (2017). Critical Behavior in Physics and Probabilistic Formal Languages. *Entropy*, 19(7):299.

McElree, B. (2000). Sentence comprehension is mediated by content-addressable memory structures. *Journal of psycholinguistic research*, 29(2):111–123.

Mikolov, T., Karafiát, M., Burget, L., Èernocký, J., and Khudanpur, Sanjeev (2010). Recurrent neural network based language model. In *Proceedings of INTERSPEECH*.

Nederhof, M. and Satta, G. (2011). Computation of infix probabilities for probabilistic context-free grammars. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1213–1221. ACL.

Neubig, G. and Mori, S. (2010). Word-based partial annotation for efficient corpus construction. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*. European Language Resources Association.

Neubig, G., Nakata, Y., and Mori, S. (2011). Pointwise prediction for robust, adaptable japanese morphological analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*, pages 529–533. The Association for Computer Linguistics.

Nicenboim, B. and Vasishth, S. (2018). Models of retrieval in sentence comprehension: A computational evaluation using bayesian hierarchical modeling. *Journal of Memory and Language*, 99:1–34.

Petrov, S. and Klein, D. (2007). Learning and inference for hierarchically split pcfgs. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 1663–1666. AAAI Press.

Roark, B. (2001). Probabilistic top-down parsing and language modeling. *Comput. Linguistics*, 27(2):249–276.

Schadeberg, T. (2003). Derivation. In *The Bantu Languages, edited by D. Nurse & G. Philippson, 71-89*. Routledge, London.

Schijndel, M. V., Exley, A., and Schuler, W. (2013). A model of language processing as hierarchic sequential prediction. *topiCS*, 5(3):522–540.

Shannon, C. E. (1951). Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Still, S. (2014). Information Bottleneck Approach to Predictive Inference. *Entropy*, 16(2):968–989.

Stolcke, A. (1995). An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Comput. Linguistics*, 21(2):165–201.

Takahashi, S. and Tanaka-Ishii, K. (2018). Cross entropy of neural language models at infinity—a new bound of the entropy rate. *Entropy*, 20(11):839.

Vaccari, O. and Vaccari, E. E. (1938). *Complete course of Japanese conversation-grammar*. Maruzen in Komm.

Vasishth, S., Nicenboim, B., Engelmann, F., and Burchert, F. (2019). Computational models of retrieval processes in sentence processing. *Trends in Cognitive Sciences*.

Xie, Z., Wang, S. I., Li, J., Lévy, D., Nie, A., Jurafsky, D., and Ng, A. Y. (2017). Data noising as smoothing in neural network language models. *arXiv preprint arXiv:1703.02573*.