Crosslinguistic Word Orders Enable an Efficient Tradeoff between Memory and Surprisal

Michael Hahn, Judith Degen, Richard Futrell

2018

Abstract

Online memory limitations are well-established as a factor impacting sentence processing and have been argued to account for crosslinguistic word order regularities. Building off expectation-based models of language processing, we provide an information-theoretic formalization of these memory limitations. We introduce the idea of a memory-surprisal tradeoff: comprehenders can achieve lower average surprisal per word at the cost of storing more information about past context. We show that the shape of the tradeoff is determined in part by word order. In particular, languages will enable more efficient tradeoffs when they exhibit information locality: when predictive information about a word is concentrated in the word's recent past. We show evidence from corpora of 52 real languages showing that languages allow for more efficient memory-surprisal tradeoffs than random baseline word order grammars.

1 Introduction

Since the 1950s, it has been a persistent suggestion that human language processing is shaped by a resource bottleneck in short-term memory. Language is produced and comprehended incrementally in a way that crucially requires both speaker and listener to use an active memory store to keep track of what was previously said. Since short-term memory of this kind is known to be highly limited in capacity (**?**), it makes sense for these capacity limits to comprise a major constraint on production and comprehension. Indeed, a great deal of work in sentence processing has focused on characterizing the effects of memory constraints on language processing (**???**).

At the same time, the field of functional linguistics has argued that these resource constraints not only affect online language processing, they also shape the form of human language itself. For example, the Performace–Grammar Correspondence Hypothesis (PGCH) of **?** holds that forms which are practically easier to produce and comprehend end up becoming part of the grammars of languages, and that this process can explain several of the universal properties of human languages originally documented by **?**.

Here we take up the question of how to characterize short-term memory capacity limitations in language processing for both speakers and listeners, and the question of whether natural language grammars are shaped by these limitations. Whereas previous theories were based on specific mechanistic models of memory, our theory is purely information-theoretic, meaning that our predictions will hold independently across a wide variety of implementations and architectures.

Our main new concept is the idea of a *memory–surprisal tradeoff*: it is possible for a listener to achieve greater ease of word-by-word comprehension at the cost of investing more computational resources into remembering previous words, and the particular shape of the resulting tradeoff of depends on the word order properties of a language. Analogous results also hold for language production by resource-constrained

speakers. We show evidence that the preferred word orders of natural languages are those that enable efficient memory–surprisal tradeoffs.

The remainder of this paper is structured as follows. In Section **??**, we give a brief review of previous work on the effects of short-term memory on languages and language processing. In Section **??**, we describe the memory–surprisal tradeoff and how it results from rate–distortion theory, the theory of optimal information processing under resource constraints (**?**). In Section **??**, We prove that word orders enable efficient processing in terms of the memory–surprisal tradeoff when they exhibit *information locality*: whenever utterance elements that predict each other are close to each other. We argue that information locality is a good model of the effects of memory constraints on language processing. In Section **??**, languages which have previously been shown to be preferred in artificial language experiments are exactly those that enable efficient memory–surprisal tradeoffs (**?**). In Section **??**, we show that word orders of natural languages as found in dependency corpora (**?**) enable more efficient memory–surprisal tradeoffs than baseline word orders. Section **??** concludes.

2 Background

{sec:backgro

A wide range of work has argued that natural language orders information in ways that reduce memory effort. An early example is **?**, who attributed the unacceptability of multiple center embeddings in English to limitations of human working memory. **?** provides cross-linguistic evidence that word orders are optimized for processing based on local contexts. Further work has found computational, corpus-based evidence that memory limitations impact language structure and production. In particular, languages have been shown to shorten the length of syntactic dependency lengths (**?**). Dependency length can be linked to memory use in certain models of incremental syntactic parsing, and increases processing difficulty in theories of memory in sentence processing (**?**). **?** further provide evidence from five languages that word orders optimize predictability from local contexts. **?** provide evidence that language shows information locality, i.e., elements with higher mutual infomation are closer together, which is predicted by their model of Lossy-Context Surprisal.

All these models of memory in sentence processing, and derived measures of efficiency for memory allocation, require specific assumptions about the architecture of memory. This leaves open the question whether such assumptions are necessary, or whether word orders across languages are optimized for memory independently of the implementation and architecture of human language processing.

We approach this question by first providing general information-theoretic lower bounds on memory load that will hold independently of the architecture of memory representations. We will consider a general setting of a listener performing incremental prediction. Our result immediatly entails a link between boundedness of memory and locality, which had been stipulated or derived from assumptions about memory architecture in previous models (**???**). We will then use corpus data from over 52 languages to provide evidence that their word orders help lower memory cost.

3 Memory-Surprisal Tradeoff

{sec:ms-trad

With minimal assumptions, we will use information theory to derive a tradeoff between *listener memory* and (listener) surprisal

In the second part of the paper, we examine whether word orders in natural language optimise this tradeoff.

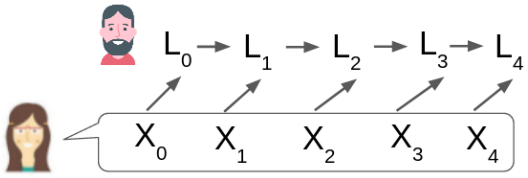

Figure 1: Illustration of (**??**). As the utterance unfolds, the listener maintains a memory state. After receiving word X_t, the listener computes their new memory state L_t based on the previous memory state L_{t-1} and the new word X_t. {fig:listene

We consider a *listener* who, as the speaker's utterance unfolds, engages in incremental prediction. For the listener, predicting the next word well requires maintaining information about the past. For the listener, the quality of prediction is measured by the average *surprisal* experienced. For a fixed language, we can ask how much information about the past (1) the speaker has to maintain to produce well-formed utterances, and (2) the listener has to maintain to incur a minimal amount of surprisal. Utilizing the tools of information theory, we quantify memory in *bits*, obtaining bounds that hold across different models of memory architecture and ways of quantifying memory load.

TODO say at some point that we're studying sentence-internal memory

3.1 Theoretical Results

We now introduce our main theoretical result on memory-surprisal tradeoffs.

Setting We formalize a language as a probabilistic sequence of words $\ldots x_{-2}x_{-1}x_0x_1x_2\ldots$, extending indefinitely both into the past and into the future. The symbols x_i belong to a common set, representing the words of the language.[1]

We model the sequence as a probabilistic sequence; that is, given a context $x_{<t}$, the next word is distributed according to a distribution $p(x_t|x_{<t})$.

We now analyze memory from the perspective of the listener, who needs to maintain information about the past to predict the future. As the speaker's utterance unfolds, the listener maintains a memory state L_t.

There are no assumptions about the memory architecture and the nature of its computations. We only make a basic assumption about the flow of information (Figure **??**): At a given point in time, the listener's memory state L_t is determined by the last word X_t, and the prior memory state L_{t-1}. As a consequence, L_t contains no information about the process beyond what is contained in the last word observed X_{t-1} and in the memory state before that word was observed L_{t-1}. This is formalized as a statement about conditional probabilities:

$$p(L_1|(X_t)_t, L_0) = p((X_t)_t|L_0, X_1) \tag{1}$$

This says that L_1 contains no information about the utterances beyond what is contained in L_0 and X_1. As a consequence, the listener has no knowledge of the speaker's state beyond the information provided in their prior communication. This is a simplification, as the listener could obtain information about the speaker from other sources, such as their common environment (weather, ...). For the study of memory in sentence processing, this seems fair. Discuss this more.

[1] Could also be phonemes, sentences, ..., any other kind of unit.

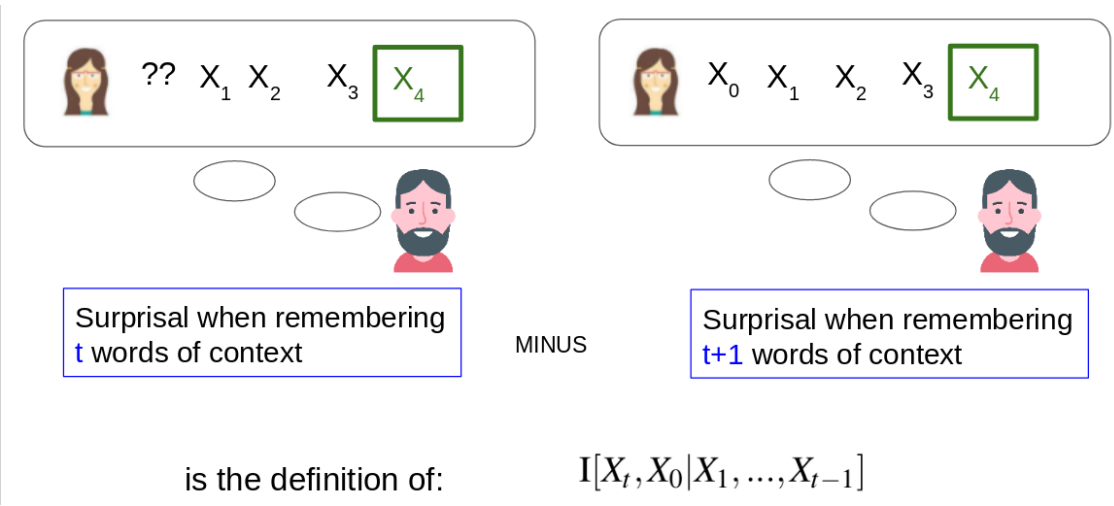

Figure 2: (TODO adapt to format) Intuitive explanation of I_t {fig:it-surp

Conditional Mutual Information We will use the concept of *conditional mutual information* (Figure **??**).

$$I_t := H[X_t|X_1,\ldots,X_{t-1}] - H[X_t|X_0,X_1,\ldots,X_{t-1}] \tag{2}$$

This is equal to the reduction in uncertainty about the t-th observation when knowing the 0-th observation, in addition to the block of intervening observations. That is, we measure the amount of statistical dependency of observations that are t steps apart, controlling for any information that is redundant with intervening observations. This quantifies how much information needs to be carried across t timesteps without any possibility for 'guessing' it from intervening observations.

We return to the information and memory curves in Figure **??**. In the graph of $t \cdot I_t$, we look for the first T such that the area under the curve to the left of T has size $\geq J$. This is illustrated in Figure **??** (right). Then let ε be the area under the curve of I_t to the right of T (Figure **??**, left). We can prove that such a listener must incur surprisal at least ε greater than a listener with perfect memory. As before, we write

$$I_t := I[X_t, X_0|X_{1\ldots t-1}]$$

Then

{prop:subopt

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

$$\sum_{t=1}^{T} tI_t \tag{3}$$

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

$$H[X_t|X_{<t}] + \sum_{t>T} I_t$$

on average.

The proof is given in the appendix (REF).

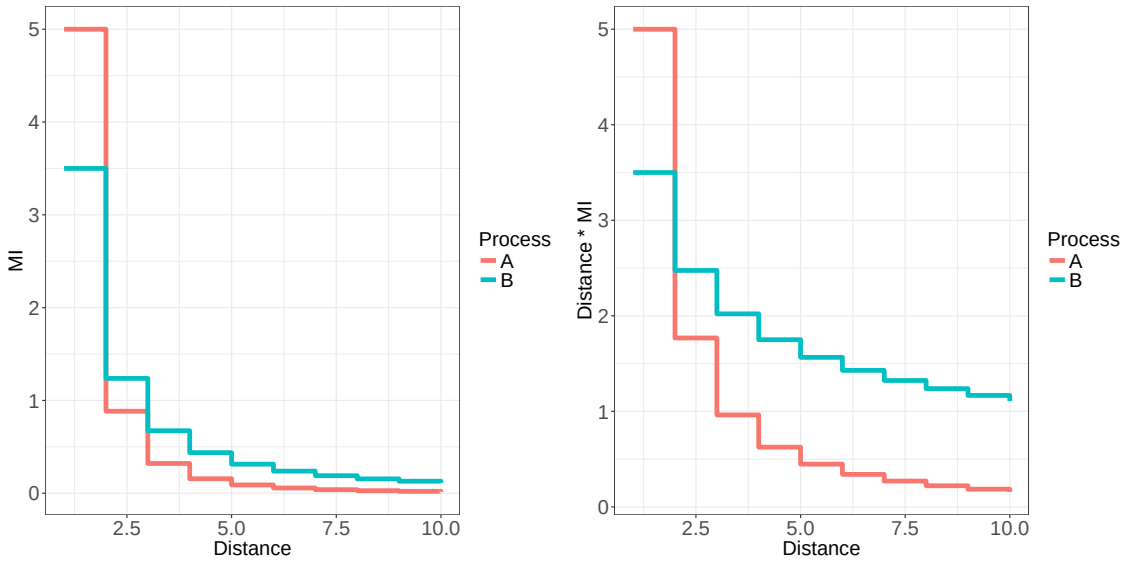

Figure 3: Left: I_t as a function of t, for two different processes. I_t decays faster for the red process: Predictive information about the present observation is concentrated more strongly in the recent past. Left: $t \cdot I_t$ as a function of t for the same processes. {fig:basic}

The proposition gives us a lower bound on the listener's memory-surprisal curve: Taking all pairs of memory $\sum_{t=1}^{T} tI_t$ and surprisal $H[X_t|X_{<t}] + \sum_{t>T} I_t$. Then interplate linearly (justify this in appendix). We obtain a curve in memry-surprisal plane, which is a lower bound on the memory demands of any listener at a given surprisal level. We visualize this for the two processes from Figure **??** in Figure **??**.

Our result is entirely information-theoretic and applies to *any* physical encoding of the past, entirely independent of the implementation of the model. In particular, while the relation to psycholinguistic and psychological models of how memory works will be interesting to explore, our result applies to any such model. Memory representations do not have to be rational or optimal for this bound to hold: It provides a *lower bound* on the amount of information that needs to be stored – other memory representations will always need to store at least as much information.

Information Locality Due to the factor t inside each term of the sum, carrying the same amount of information over longer distances requires more memory – that is, modeling long statistical dependencies is more costly in terms of memory than modeling shorter ones. This formalizes a general, assumption-free, link between memory and locality in language production. In Section **??**, we will extend this analysis to listeners performing incremental prediction.

The proposition implies that memory is decreased if I_t decreases quickly as $t \to \infty$ – that is, if the contributions of long-term dependencies in the process are small. In particular, memory load can only be finite if I_t decreases fast enough for the infinite sum to converge to a finite value.

We illustrate Proposition **??** in Figure **??**. We consider two processes A and B, where $I_t := 5t^{-1.5}$ for A and $I_t := 3.5t^{-2.5}$ for B. The curves of I_t, as a function of the distance t, are shown in Figure **??** (left). In both cases, I_t converges to zero as t grows to infinity. However, I_t decays more quickly for Process A (red). This means that predictive information about an observation is concentrated more strongly in the recent past. In Figure **??** (right), we show $t \cdot I_t$ as a function of t. Note that the area under the curve is equal to (**??**). This

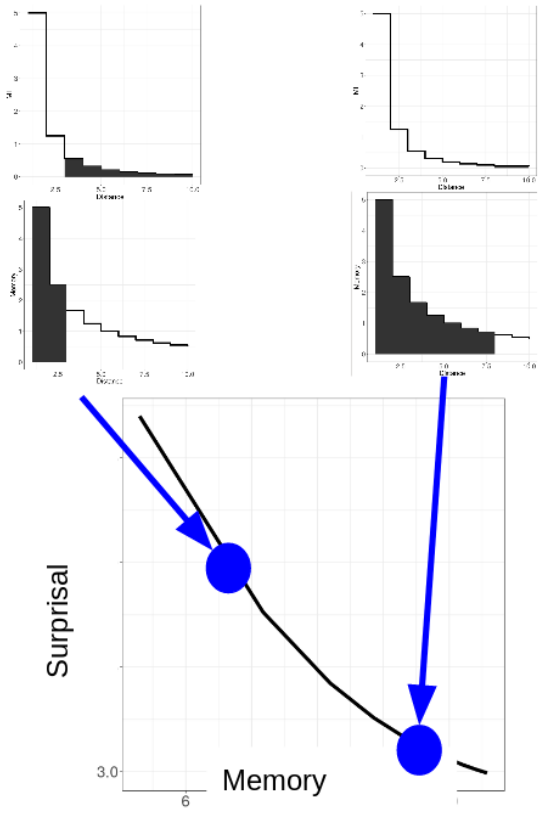

Figure 4: Estimating memory-surprisal tradeoff using the Theorem: We trace out the memory and surprisal values for all $T = 1, 2, ...,$ and linearly interpolate the curve. {fig:interpo

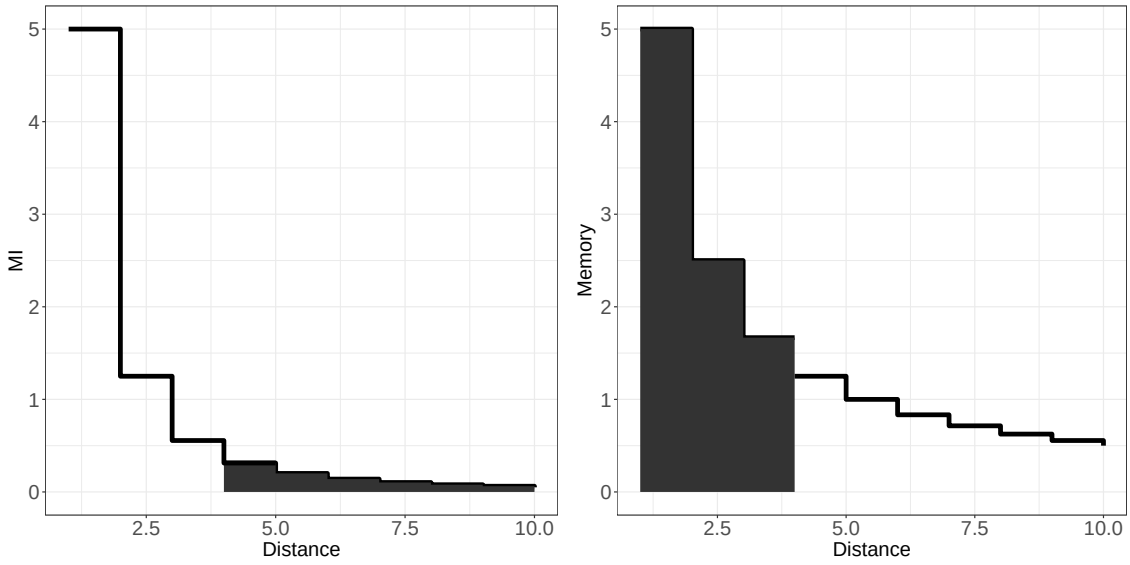

Figure 5: Illustration for Proposition **??**. Listeners can trade off memory and surprisal: A listener only investing memory of the amount given by the black area on the right will incur at least the black area on the left in additional surprisal. In the given example, $T = 4$. By varying T, the two areas describe the listener's memory-surprisal tradeoff curve. {fig:listene

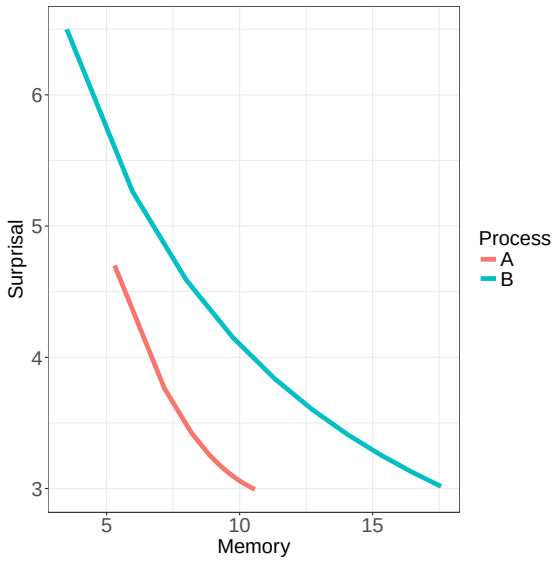

Figure 6: Listener's memory-surprisal tradeoff for the two processes in Figure **??**. Recall that the red process had a faster decay of conditional mutual information. Correspondingly, this figure shows that a listener can achieve lower surprisal at the same level of memory load. {fig:listene

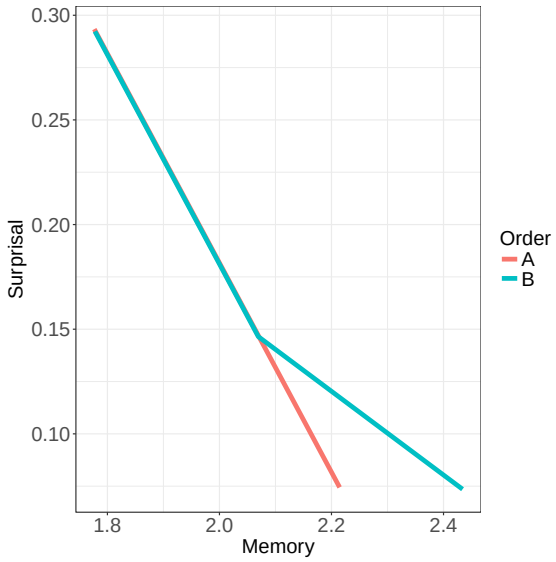

Figure 7: Tradeoff between listener memory and surprisal, for the two versions of the artificial language from **?**. Language A requires less memory at the same level of surprisal. {fig:toy-lis

area is smaller for the red process, as I_t decays more quickly there.

4 Experiment 1: Memory and Dependency Length

We illustrate the linguistic predictions of Proposition **??** by reanalyzing the data from **?**. This is a miniature artificial language study that showed a bias for Dependency Length Minimization in production in artificial language learning. Due to the controlled setting, it is possible to exactly compute the speaker's memory as given in (**??**).

As (**??**) is invariant under reversal of the language, we only consider the head-final version of her artificial language. The language has consistent head-final order, and uses case marking on objects. The relevant production targets are transitive sentences where one of the two arguments is much longer than the other, due to the presence of a PP modifier, as shown in Table **??**. The language has variable order of subjects and objects; for the production targets, the B versions produce much longer dependencies than the A versions. Dependency Length Minimization thus predicts that speakers are more likely to use the A versions. **?** confirmed this experimentally.

In this section, we show that our bound on speaker memory makes the same prediction, without reference to syntactic structure or specific memory architectures.

We constructed one language consisting of the A versions, and one language consisting of the B versions. Following the experimental setup of **?**, we assigned equal probability to the two possible configurations per language, and used a separate set of nouns (inanimate nouns) for the embedded noun in the long phrase.

We interpreted each of the two languages as a stationary processes, extending infinitely in both directions, by concatenating independent samples drawn from the language. We computed (**??**) from a chain of 1000 independently sampled sentences, for each of the two versions of the toy language. Figure **??** (left) shows the curve of the conditional mutual information I_t as a function of the distance t. The curves differ at $t = 2$ and $t = 5$: About 0.073 nats of predictive information that are at distance $t = 2$ in the A orders are

A Orders: Short Dependencies

Object				Subject	Verb
Adjective	Noun	Postposition	Noun-Case	Noun	Verb

Subject				Object	Verb
Adjective	Noun	Postposition	Noun	Noun-Case	Verb

B Orders: Long Dependencies

Subject	Object			Verb
Noun	Adjective	Noun	Postposition	Noun-Case	Verb

Object	Subject				Verb
Noun-Case	Adjective	Noun	Postposition	Noun	Verb

Table 1: Production targets in the artificial mini language from **?**. The language has head-final order, with free variation between SO and OS orders. When one of the arguments is much longer than the other, placing the longer one first ('A' orders) shortens syntactic dependencies, compared to 'B' orders. {tab:artific

moved to $t = 5$ in the B orders. The source of the difference lies in predicting the presence and absence of a case marker on the second argument – i.e., whether to anticipate a subject or object. In the A orders, considering the last two words is sufficient to make this decision. In the B orders, it is necessary to consider the word before the long second constituent, which is five words in the past.

The total amounts of predictive information – corresponding to the area under the curve – are the same, indicating that both languages are equally predictable. However, we will see that the memory demands are different: Figure **??** (right) shows $t \cdot I_t$ as a function of t. As I_t decays faster in A orders, the total area under the curve now differs between A and B, and is larger in B.

In Figure **??**, we show the resulting curve for the two versions of the artificial language from **?**. The curve shows that, at any desired level of surprisal, Order A requires at most as much memory as Order B. For reaching optimal surprisal, Order A requires strictly less memory. Thus, in this case, the listener's surprisal-memory tradeoff is optimized by the orders predicted by Dependency Length Minimization.

It is important to stress that, even though we computed this value by considering the number of words impacting predictions at a given point in time, this bound holds independently of the actual implementation and architecture of memory and predictions.

Center Embeddings **?** attributed the unacceptability of multiple center-embedding to memory limitations.

Other Psycholinguistic Predictions

Speakers

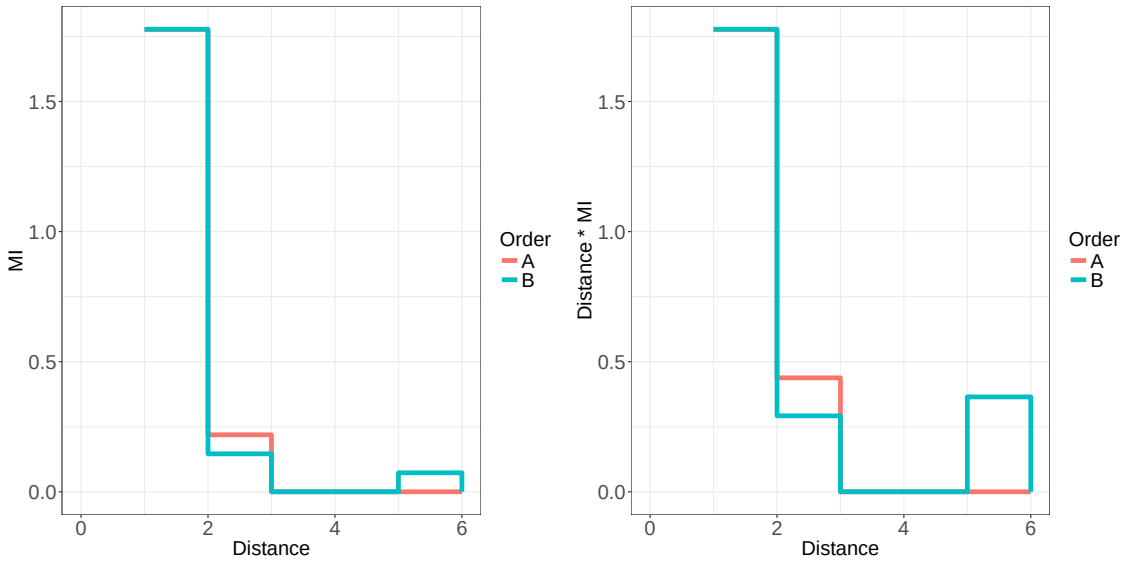

Figure 8: Left: Decay of Conditional Mutual Information, as a function of the distance t, for the two versions in the artificial language. The areas under the two curves are identical, corresponding to the fact that both orders are equally predictable. However, mutual Information decays faster in Order A. Right: tI_t, as a function of t. The area under the B curve is larger, corresponding to larger memory demand for this order. {fig:toy-mis

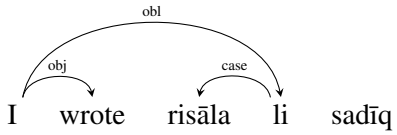

Figure 9: TODO Dependencies example {fig:depende

5 Large-Scale Evidence that natural language optimize Memory-Surprisal Tradeoff

We now investigate whether word orders as found in natural language optimize the two memory-surprisal tradeoffs. We compare the memory-surprisal tradeoffs of 52 actual languages to those of counterfactual reorderings. We cannot just compare to random orderings of individual syntactic trees, as such languages would not have word order regularities. Therefore, we compare to counterfactual word order grammars.

5.1 Data

We draw on corpora annotated with syntactic structures. The Universal Dependencies project has compiled dependency corpora for several dozen languages (**?**).

Dependency Grammar In dependency corpora, sentences are annotated with *dependency trees* (Figure **??**). These are directed trees describing the grammatical relations among words. For example, the arcs labeled "obj" represent that the noun in question is the *direct object* if the verb, rather than e.g. the subject or an indirect object. A dependency arc is drawn from a *head* (e.g. TODO in Figure TODO) to a *dependent* (e.g.

TODO). Dependency trees can be defined in terms of many different syntactic theories **?**. Although there are some differences in how different formalisms would draw trees for certain sentences, there is broad enough agreement about dependency trees that it has been possible to develop large-scale dependency-annotated corpora of text from dozens of languages **?**.

Corpora We considered all languages for which there are Universal Dependencies 2.2 treebanks with a total of at least 500 sentences of training data. We excluded data from historical languages.[2] This resulted in 52 languages.

For each of these languages, we pooled all available corpora in one dataset. We excluded corpora that primarily contain text created by non-native speakers. Universal Dependencies corpora have a predefined split into *training*, *held-out* (also known as *development*), and *test* partitions. While larger corpora have all three partitions, smaller corpora often have only some of these partitions. For most language, we used the predefined data split, separately pooling data from the different partitions. For some languages with little data, there is no predefined training partition, or the training partition is smaller than the other partitions. In these cases, we redefined the split to obtain more training data: For these languages, we pooled all the available partitions, used 100 randomly selected sentences as held-out data, and used the remainder as training data.[3] For each language, we used the training and held-out sets for estimating the memory-surprisal tradeoff (see Section **??**). We provide the sizes of the resulting datasets in Table **??**.

5.2 Counterfactual Ordering Grammars

We define ordering grammars, small models of the rules by which languages order syntactic structures into sentences. Our formalism of ordering grammars adapts the method of **???** to the setting of dependency corpora.

Universal Dependencies defines 37 universal syntactic relations that are used to label dependency arcs across all corpora. These relations encode cross-linguistically meaningful relations such as subjects, objects, and adjectival modifiers. We define ordering grammars by assigning a parameter $a_\tau \in [-1, 1]$ to every one of these 37 universal syntactic relations. Relations sometimes have language-specific subtypes; we do not distinguish these subtypes.

In our model, this parameter defines how dependents are ordered relative to their head: Given a head and a set of dependents, we order each dependents by the parameter a_τ assigned to the syntactic relation linking it to the head. Dependents with negative weights are placed to the left of the head; dependents with positive weights are placed to the right.

Ordering grammars describe languages that have consistent word order: For instance, the subject is consistently ordered before or after the verb, depending on whether the parameter for the verb-subject dependency is positive or negative.

We define baseline grammars by randomly sampling the parameters a_τ. Such baseline grammars define languages that have consistent word order, but do not exhibit any systematic correlations between the orderings of different dependents.

Discussion In actual languages, the ordering of words is largely determined by the syntactic relations (CITE). However, certain kinds of rules cannot be modeled by our word order grammars, such as rules sensitive to the category of the dependent (e.g., differences between nominal and pronominal objects). Word

[2] Ancient Greek, Coptic, Gothic, Latin, Old Church Slavonic, Old French.

[3] This affects Amharic, Armenian, Breton, Buryat, Cantonese, Faroese, Kazakh, Kurmanji, Naija, Thai, and Uyghur.

order freedom also is not modeled. In this sense, ordering grammars represent approximations to the kinds of ordering rules found in natural language **???**.

5.3 Estimating Memory-Surprisal Tradeoff

{sec:method}

To estimate mutual informations, we use LSTM recurrent neural networks, the basis of the state of the art in modeling natural language (CITE) and predicting the surprisal effect on reading times (**??**). We provide data from alternative estimation methods in the SI.

Model and Parameter Estimation We use a recurrent neural network with Long-Short-Term Memory cells (**?**) (CITE for Neural LM). This architecture takes as input a sequence $x_1 ... x_N$ of words, and at each time step $t = 1, ..., N$, calculates a probability distribution over the next word w_t given preceding words $w_1 ... w_{t-1}$: $p(w_t | w_1 ... w_{t-1})$.

The network is parameterized by a vector θ of weights determining how the activations of neurons propagate through the network (**?**). 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.

The training corpus is chopped into word sequences $w_1 ... w_T$ of length T ($T = 20$ in our experiments). If θ_n consists of the LSTM parameters after n training steps, we randomly select a word sequence $w_1 ... w_T$ from the training corpus, and use the LSTM using the current parameter setting θ_n to compute the per-word surprisals. We then update the parameter vector:

$$\theta_{n+1} := \theta_n + \alpha \partial_\theta \left(\sum_{i=1}^{T} \log p_\theta(w_i | w_1 ... w_{i-1}) \right) \tag{4}$$

where $\alpha \in \mathbb{R}_+$ is the *learning rate*. When calculating the parameter update, we use three standard methods of regularization that have been shown to improve neural language modeling: dropout (**?**), word dropout, and word noising (**?**). In this process, the word sequences are sampled without replacement. 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 at the current parameter setting θ_n 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.

Choice of Hyperparameters The LSTM model has a set of numerical *hyper-parameters* that need to be specified before parameter estimation, namely the dimensionalities of the embeddings d_{emb}, the dimensionality of the hidden states d_{LSTM}, and the number of LSTM layers d_{layer}, the learning rate α, and the regularization parameters (dropout rate $p_{embedding}$, word dropout rate p_{word}, word noising rate $p_{noising}$). We choose these parameters so as to minimize the average surprisal on the held-out partition resulting at the end of parameter estimation.

For each corpus, we used Bayesian optimization using the Expected Improvement acquisition function (**?**) to find a good setting of the hyperparameters. We optimized the hyperparameters to minimize average surprisal on languages generated from random word order grammars. This biases the hyperparameters towards modeling counterfactual grammars better, biasing them *against* our hypothesis.

For computational efficiency, neural language models can only process a bounded number of distinct words in a single language. 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 (CITE). Following (CITE), 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 this 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 (see SI).

Estimating the Memory-Surprisal Tradeoff Curve The quantity $\mathrm{I}[X_t, X_0 | X_1, ..., X_{t-1}]$ in (**??**) is equal to the difference

$$H[X_t|X_1, ..., X_{t-1}] - H[X_t|X_0, X_1, ..., X_{t-1}] \tag{5}$$

For each word in the held-out partition, we compute the difference

$$-\log P_\theta[X_t|X_0, X_1, ..., X_{t-1}] - P_\theta[X_t|X_1, ..., X_{t-1}] \tag{6}$$

and take the average over these. We cut T off at 20, as this is the length of the sequences processed by the model.

We then used linear interpolation to interpolate the surprisal value for memory values in between these values. (TODO make a figure). This is justified theoretically (TODO maybe discuss when introducing the theorem).

We estimate the unigram entropy $H[X_0]$ by averaging over all models.

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 (REF).

5.4 Statistics

We now describe how we compared memory-surprisal tradeoffs between real and baseline languages.

We want to test whether languages' surprisal-memory tradeoffs better than those of most baseline languages. We compare real and baseline languages by evaluating which languages result in lower surprisal at the same level of memory. We now describe the statistics we use to quantifying the difference between real and baseline languages. We do everything in a frequentist framework (null hypothesis testing & confidence intervals), as we can do exact tests and confidence intervals without parametric assumptions. Maybe we can explain how the tests & CIs also have reasonable Bayesian interpretations (for the specific methods used here, rejection of the null should guarantee that the posterior of the null hypothesis is small under a wide range of priors.).

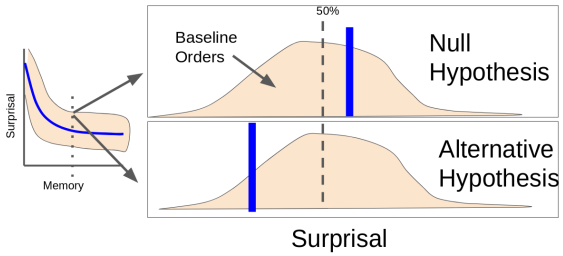

Figure 10: Illustration for the pointwise null-hypothesis significance test. At a given level of memory, we test against the null hypothesis that at least half of the baseline orders provide lower surprisal than the real language. {fig:nhst-po

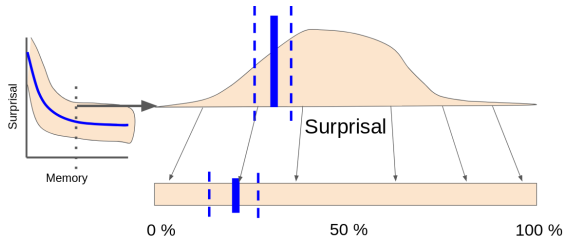

Figure 11: Illustration for the quantile estimate. At each level of memory, we provide an estimate of the percentage of baseline languages that have lower surprisal than the real language. {fig:quantil

Confidence Interval for Medians We use a (nonparametric and nonasymptotic) confidence interval for the median surprisal at each memory value, using the binomial test. We consider the medians over all runs for the real language, and over all baselines grammars.

CI for Median Difference We create (nonparametric and nonasymptotic) confidence interval for the difference between real and baseline median surprisals at each memory value.

Pointwise Significance Test For each memory value μ, we do a nonparametric and nonasymptotic significance hypothesis test against the null hypothesis that at least half of the baseline grammars have lower surprisal than the actual language (Figure **??**). Formally, let $W_{-}(\mu)$ be the proportion of baseline languages that have strictly lower surprisal than the real language at memory level μ. We take the real language to be represented by the *sample median*. For each level μ of memory, we consider the null hypothesis that

$$W_{-}(\mu) \leq 0.5 \tag{7}$$

We use the Binomial Test.

Pointwise Quantile Estimate For each level of memory, we estimate what percentage of baseline languages have lower surprisal than the real language. This is described in Figure **??**.

We derive a confidence interval under the assumption that the distribution of baseline languages is unimodal.

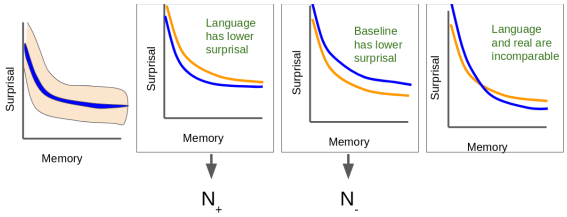

Figure 12: Illustration for the global quantile estimate. For each sample for the real language, we compare the memory-surprisal curve to all baselines. {fig:quantil

We derive a CI for the quantile, under the assumption that the baseline distribution is unimodal. We take the REAL values to be estimated exactly by their medians.

We want to create a CI at each Memory value for the quantile.

Let n_+ be the better baseline samples, n_- the worse ones.

Let θ the best baseline sample that is worse than the real median.

We want to get a confidence bound q on $P(X < x_{real})$.

Let $p := P(N_+ \leq n_+ | N_+ + N_-; q)$. Then output $(0, q)$ as a level p CI for the parameter $P(X < x_{real})$.

We minimize q subject to $p < 0.05$.

Once we have q, we can make it a bit better under the assumption that the baseline distribution is unimodal.

This CI is exact in the sense that it does not involve asymptotic approximations or parametric assumptions, but it is extremely conservative.

Also the following does not assume unimodality, and ends up getting about the same intervals

Global Quantile Estimate 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

$$G := \frac{\mathbb{E}_{x \sim P_1}[W_+(x)]}{\mathbb{E}_{x \sim P_1}[W_+(x) + W_-(x)]} \tag{8}$$

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.

Unlike the other statistics, this one provides a global measure of the degree of optimization of the real language. Due to the use of bootstrapping, the confidence intervals are not exact.

5.5 Number of Samples

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 (CITE).

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