

# 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

June 24, 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 (?). 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  $\dots w_{-2}w_{-1}w_0w_1w_2\dots$  (?). 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* (?). 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  $\Delta$  of time steps. That is, for any integers  $t$ ,  $\Delta$ , and  $T > 0$ :

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

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 (???), excess entropy (??), and mutual information (??).

### 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  $m_t$ :

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

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  $t$ ,  $\Delta$ , and  $T > 0$ :

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

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

$$S_M \equiv H[w_t|m_t]. \quad (6)$$

and memory cost

$$H_M \equiv H[m_t] \quad (7)$$

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:

$$m_t \perp w_{\geq t} | w_{<t} \quad (8)$$

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  $S_\infty$  are given by

$$S_M \equiv H[w_t | m_t] \quad (9)$$

$$S_\infty \equiv H[w_t | w_{<t}] \quad (10)$$

We restate the theorem:

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

$$H_M \leq \sum_{t=1}^T t I_t \quad (11)$$

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

$$S_M \geq S_\infty + \sum_{t>T} I_t \quad (12)$$

*on average.*

*Proof.* The difference between the listener’s average surprisal  $S_M$  and optimal surprisal  $S_\infty$  is

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

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

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

Due to Processing Postulate 1, we have

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

and therefore the Data Processing Inequality (?) entails the following inequality for every positive integer  $t$ :

$$H[w_t | m_t] \geq H[w_t | w_{1..t-1}, m_1]. \quad (16)$$

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:

$$H[w_t | m_t] - H[w_t | w_{<t}] \geq \frac{1}{T} \sum_{t'=1}^T (H[w_{t'} | w_{1..t'-1}, m_1] - H[w_{t'} | w_{1..t'-1}, w_{\leq 0}]) \quad (17)$$

$$= \frac{1}{T} (H[w_{1..T} | m_1] - H[w_{1..T} | w_{\leq 0}]) \quad (18)$$

$$= \frac{1}{T} (I[w_{1..T} : w_{\leq 0}] - I[w_{1..T} : m_1]). \quad (19)$$

The first term  $I[w_{1..T} : w_{\leq 0}]$  can be rewritten in terms of  $I_t$  using the chain rule of mutual information (?):

$$I[w_{1..T} : w_{\leq 0}] = \sum_{i=1}^T \sum_{j=-1}^{-\infty} I[w_i : w_j | w_{j+1} \dots w_{i-1}] = \sum_{t=1}^T t I_t + T \sum_{t>T} I_t. \quad (20)$$

Therefore

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

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

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

Rearranging yields

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

as claimed.  $\square$

### 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 (???). We extend our model of memory in incremental prediction to capture key aspects of the models described by ???.

The ACT-R model of ? 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:

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

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

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

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

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

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

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

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

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 ?, 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 ?, a retrieval operation takes longer if more chunks are similar to the retrieval cue, whereas, in the direct-access model (???), 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 ? 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  $H[m_t]$  is bounded as*

$$H[m_t] \leq \sum_{t=1}^T tI_t \quad (28)$$

*and the average amount of retrieved information is bounded as*

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

*Then the surprisal cost is lower-bounded as*

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

*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  $w_{1...t}, m_0, r_0, \dots, r_t$ . Therefore, the Data Processing Inequality entails:

$$H[w_t|m_{t-1}, r_t] \geq H[w_t|w_{1...t}, m_0, r_0, \dots, r_t]. \quad (31)$$

As in (17), this leads to

$$H[w_t|m_{t-1}, r_t] - H[w_t|w_{<t}] \geq \frac{1}{T} \sum_{t=1}^T (H[w_t|w_{1...t}, m_0, r_0, \dots, r_t] - H[w_t|w_{1...t-1}, w_{\leq 0}]) \quad (32)$$

$$\geq \frac{1}{T} (H[w_{1...T}|m_0, r_0, \dots, r_T] - H[w_{1...T}|w_{\leq 0}]) \quad (33)$$

$$= \frac{1}{T} (I[w_{1...T}, w_{\leq 0}] - I[w_{1...T}, (m_0, r_0, \dots, r_T)]). \quad (34)$$

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

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

Due to the inequalities

$$I[w_{1...T}, m_0] \leq H[m_0] \leq \sum_{t=1}^T tI_t \quad (35)$$

$$I[w_{1...T}, r_t | m_0, r_{1...t-1}] \leq H[r_t] \leq \sum_{t=T+1}^S I_t, \quad (36)$$

this can be bounded as

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

Finally, this reduces as

$$H[w_t | m_{t-1}, r_t] - H[w_t | w_{<t}] \geq \frac{1}{T} (T \sum_{t>T} I_t - T \cdot H[r_t]) \quad (38)$$

$$= \sum_{t>T} I_t - H[r_t] \quad (39)$$

$$\geq \sum_{t>T} I_t - \sum_{t=T+1}^S I_t \quad (40)$$

$$= \sum_{t>S} I_t. \quad (41)$$

□

**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

$$\lambda_1 H[m_t] + \lambda_2 H[r_t]. \quad (42)$$

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

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

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 (?). 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 (?), 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  $q$ :

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

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

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  $q$ :

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

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

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 probability 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:

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

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

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

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*

$$H[m_t] \leq \sum_{i=1}^T t I_t, \quad (50)$$

then

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

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

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

*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

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

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

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

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

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

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

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.  $\square$

## 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:

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

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  $u_{t+1}^{(i)} = 1$ , 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.<sup>1</sup>

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  $v_{t-2k+1}, \dots, v_t$ , 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.

<sup>1</sup>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  $I[Z, W] = D_{KL}(p(z|w=1)||p(z)) = \frac{1+l}{2l} \log \frac{1+l}{1/2} + \frac{l-1}{2l} \log \frac{l-1}{1/2} = \frac{1+l}{2l} \log \frac{1+l}{1} + \frac{l-1}{2l} \log \frac{l-1}{1} \leq \frac{1+l}{1} \log \frac{1+l}{1} = (1 + 1/l) \log(1 + 1/l) \leq (1 + 1/l)(1/l) = 1/l + 1/l^2 < 2/l = \frac{1}{2k}$ .

## 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  $x_{t+1}, x_{t+2}, \dots$  optimally from the past requires storing  $y_{t-2k+1}, \dots, y_t$  and  $z_t$ . This amounts to

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

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

We now evaluate  $I_t$ . We have

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

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

and all other values of  $I_t$  are zero.

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

$$H_M \geq \sum_{t=1}^{\infty} t I_t = 2k I_{2k} = 1 \quad (62)$$

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

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

$$m_{t+1} := z_t \quad (64)$$

Then, if  $z_t = 1$ , we retrieve

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

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  $S \rightarrow \infty$ , we obtain

$$H[m_t] \geq I_1 = H_2[1/4k^2] \quad (66)$$

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

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  $\lim_{T \rightarrow \infty} \sum_{t=1}^T t I_t = 1$ .

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 ?). 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  $\lim_{T \rightarrow \infty} \sum_{t=1}^T t I_t > 0$  bits of memory (?).

## 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  $\theta$  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  $\theta$  are randomly initialized to some setting  $\theta_0$ .

The training corpus is chopped into word sequences  $w_1 \dots w_{T_{max}}$  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  $\theta$  so as to minimize the surprisal

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

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 (?).

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  $\theta$  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  $p(w_t | w_1 \dots w_{t-1})$  are dominated by a small number of preceding words (?), 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 (?) to find a good setting of the hyperparameters. We optimized the hyperparameters to minimize average surprisal (??) 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 (?). 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:

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

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

$$\widehat{S}_T = \frac{1}{|HeldOut| - T} \sum_{i=T}^{|HeldOut|} \log P_{\theta}[w_i | w_{i-T}, w_{i-T+1}, \dots, w_{i-1}] \quad (69)$$

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

$$S_t \approx \min_{s \leq t} \widehat{S}_s, \quad (70)$$

which amounts to only considering higher-order models  $P_{\theta}[w_t | w_{t-T}, w_{t-T+1}, \dots, w_{t-1}]$  when they improve over lower-order ones. This procedure ensures that  $\widehat{S}_t$  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 ??.

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

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

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  $\leq 0.15$ , 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.<sup>2</sup>

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.

### 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

<sup>2</sup>Due to a scripting error, a much higher number of samples was generated for Erzya.



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