On the Proper Treatment of Units in Surprisal Theory

Samuel Kiegeland[Σ,Δ] **Vésteinn Snæbjarnarson**[Σ,U] **Tim Vieira**[Σ] **Ryan Cotterell**[Σ]
[Σ]ETH Zürich [Δ]CHI-FRO [U]University of Copenhagen
{samuel.kiegeland, vest.snae, tim.f.vieira}@gmail.com
ryan.cotterell@inf.ethz.ch

Abstract

Surprisal theory links human processing effort to the predictability of an upcoming linguistic unit, but empirical work often leaves the notion of a *unit* underspecified. In practice, experimental stimuli are segmented into linguistically motivated units (e.g., words), while pretrained language models assign probability mass to a fixed token alphabet that typically does not align with those units. As a result, surprisal-based predictors depend implicitly on ad hoc procedures that conflate two distinct modeling choices: the definition of the unit of analysis and the choice of regions of interest over which predictions are evaluated. In this paper, we disentangle these choices and give a unified framework for reasoning about surprisal over arbitrary unit inventories. We argue that surprisal-based analyses should make these choices explicit and treat tokenization as an implementation detail rather than a scientific primitive.

https://github.com/samuki/units-surprisal

1 Introduction

A long line of work in psycholinguistics has sought to characterize the processing difficulty that a comprehender experiences upon encountering a linguistic unit in context (Miller and McKean, 1964; Ehrlich and Rayner, 1981; Balota et al., 1985, *inter alia*). A prominent *computational* (Marr, 1982) account of such processing difficulty is surprisal theory (Hale, 2001; Levy, 2008), which posits that processing effort is determined by a unit's surprisal: the negative log-probability of encountering that unit given its preceding context.[1]

In early experimental work on surprisal theory, researchers often built and trained their own language models for a given dataset and experimental paradigm (Hale, 2001; Levy, 2008; Demberg and Keller, 2008; Mitchell et al., 2010; Goodkind and Bicknell, 2018, *inter alia*). Because they controlled the entire modeling pipeline, they were free to choose the basic units of the language model, i.e., its *alphabet*. For example, in his seminal work, Hale (2001) trained a probabilistic context-free grammar over units derived from the Penn Treebank (Marcus et al., 1993), i.e., units that follow the Penn Treebank's tokenization scheme.[2] However, as language models grew—both in parameter count and in the size of their training corpora—it became inconvenient, if not infeasible, to train models from scratch for each study using the proper alphabet.

With the shift toward large pretrained models (Wilcox et al., 2020, 2023; Oh and Schuler, 2023), experimenters are no longer free to attune the model's alphabet to their datasets. Instead, they inherit a fixed tokenization, e.g., byte-pair encoding (Gage, 1994; Sennrich et al., 2016), whose units

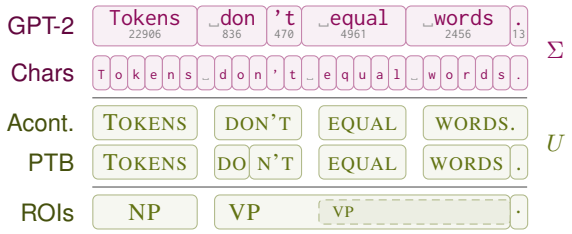

Figure 1: The string *Tokens don't equal words.* at three levels: two alphabets of symbols Σ, two unit inventories U, and regions of interest (ROIs) derived from the sentence's constituency parse: NP, VP (with a nested inner VP shown dashed), and punctuation. The contraction *don't* is split three ways: GPT-2 yields don | 't, Penn Treebank (PTB) yields DO | N'T, and the acontextual inventory keeps it as one unit DON'T. The period in WORDS. is similarly attached in the acontextual inventory but separate in the contextual one.

[1]This probability is understood as the comprehender's own predictive distribution over upcoming linguistic units, derived from an unobserved human language model. Empirical studies typically approximate this distribution with language models trained on natural language text.

[2]In line with the fashion of the time, Hale (2001) populated the model's vocabulary with high-frequency words from the training portion of the Penn Treebank, together with a distinguished out-of-vocabulary symbol.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 32202–32224
July 2-7, 2026

generally do not coincide with linguistically meaningful ones (Church, 2020; Hofmann et al., 2021; Nair and Resnik, 2023). This mismatch has given rise to a methodological infelicity. Researchers must reconcile the gap between their desired units and the model's alphabet (see Figure 1), typically through bespoke post hoc procedures that impose unit boundaries on token strings (Wilcox et al., 2020; Nair and Resnik, 2023; Wilcox et al., 2023; Pimentel and Meister, 2024; Oh and Schuler, 2024). Such heuristics vary widely between papers and muddle the interpretation of the units.

Units are not the only level at which we may seek an analysis. For instance, one may wish to relate surprisal to discourse structure (Tsipidi et al., 2024, 2025) by aggregating word-level surprisals into a discourse-level region of interest (ROI; Giulianelli et al., 2024). Beyond the choice of units, then, the modeler must also determine the ROIs, i.e., how units are aggregated into predictors. Units and ROIs often coincide one-to-one, but need not: ROIs may span multiple units or even overlap, unlike units themselves. One might *a priori* construct a language model with discourse-level units, but this requires a more permissive definition than the one given here, because discourse structures are nested like syntactic constituents. See Figure 1 for an example of nested constituent ROIs.

This paper seeks to promote an understanding of the proper treatment of units and ROIs in surprisal theory, and provides a practical toolkit that enables modelers to select their desired units freely. Our proposal is straightforward: the experimenter first chooses the unit inventory best suited for their analysis. Then, if the language model cannot be retrained, it should be converted to the chosen alphabet, e.g., by composing the language model with an appropriate function (Snæbjarnarson et al., 2026). On this view, tokenization is little more than an implementation detail that should be of no scientific importance; the unit of analysis is a modeling choice and should be selected to match the scientific question at hand. Depending on the goal, units may be taken to be the model's *token* alphabet (Beinborn and Pinter, 2023; Nair and Resnik, 2023), they can be defined by an explicit segmentation scheme, such as simple delimiter rules (Oh and Schuler, 2024; Pimentel and Meister, 2024), or even derived using contextual segmentation rules, such as the Penn Treebank guidelines (PTB; Marcus et al., 1993) or Universal Dependencies (UD; Nivre et al., 2017).

2 Language Models

Let Σ be an **alphabet**, i.e., a finite, non-empty set whose elements are called **symbols**. A **string** over Σ is a finite sequence of symbols $\boldsymbol{\sigma} = \sigma_1\sigma_2\cdots\sigma_T$ with $\sigma_t \in \Sigma$. We write Σ^* for the set of all strings over Σ, including the **empty string** ε, and $\Sigma^+ \stackrel{\text{def}}{=} \Sigma^* \setminus \{\varepsilon\}$. Furthermore, we write $\text{EOS} \notin \Sigma$ for a distinguished **end-of-sequence** symbol. We use $\boldsymbol{\sigma}\cdot\boldsymbol{\sigma}'$ to denote the **concatenation** of $\boldsymbol{\sigma}, \boldsymbol{\sigma}' \in \Sigma^*$ and write $\boldsymbol{\sigma} \preceq \boldsymbol{\sigma}'$ if $\boldsymbol{\sigma}$ is a prefix of $\boldsymbol{\sigma}'$.

A **language model** p_Σ is a probability distribution over Σ^*. For any $\boldsymbol{\sigma} \in \Sigma^*$, $p_\Sigma(\boldsymbol{\sigma})$ factors as

$$p_\Sigma(\boldsymbol{\sigma}) = \overrightarrow{p}(\text{EOS} \mid \boldsymbol{\sigma}) \prod_{t=1}^{T} \overrightarrow{p}(\sigma_t \mid \boldsymbol{\sigma}_{<t}), \quad (1)$$

where the **conditional prefix probability** is

$$\overrightarrow{p}(\boldsymbol{\sigma}' \mid \boldsymbol{\sigma}) \stackrel{\text{def}}{=} \Pr_{Y \sim p_\Sigma}\left[Y \succeq \boldsymbol{\sigma}\cdot\boldsymbol{\sigma}' \,\middle|\, Y \succeq \boldsymbol{\sigma}\right] \quad (2)$$

$$= \frac{\overrightarrow{p}(\boldsymbol{\sigma}\cdot\boldsymbol{\sigma}')}{\overrightarrow{p}(\boldsymbol{\sigma})}, \quad (3)$$

and the **prefix probability** is

$$\overrightarrow{p}(\boldsymbol{\sigma}) \stackrel{\text{def}}{=} \Pr_{Y \sim p_\Sigma}\left[Y \succeq \boldsymbol{\sigma}\right] \quad (4)$$

$$= \sum_{\boldsymbol{\sigma}' \in \Sigma^*} \mathbb{1}\{\boldsymbol{\sigma}' \succeq \boldsymbol{\sigma}\}\, p_\Sigma(\boldsymbol{\sigma}'). \quad (5)$$

3 Units

From the cognitive perspective, linguistic utterances are generally taken to be divisible into discrete units.[3] The question of what constitutes a linguistic unit has been debated at least since de Saussure (1997) and Bloomfield (1933), and remains contentious to this day (Haspelmath, 2011; Murphy, 2024). Informally, a unit is a discrete segment of linguistic structure—a phoneme, morpheme, word, or phrase—that serves as an atom of analysis at a chosen level of description. It is generally agreed that linguistic units exist at various levels, i.e., utterances can be divided into clauses, clauses into words, words into morphemes, morphemes into phonemes, and phonemes into phones—even while the notion of a word is a hotly contested one (Haspelmath, 2011; Dixon and Aikhenvald, 2002). Each of these levels provides a valid granularity at

[3]The claim that speech is divisible into discrete units is itself an idealization. The continuous acoustic signal is carved into discrete segments by convention; the mapping from articulatory gestures to perceived phonemes involves substantial coarticulation and context-dependence (Liberman et al., 1967).

which one can decompose an utterance (Hockett, 1958). The choice of level is not merely a technical nuisance but a substantive commitment about the granularity at which cognitive processing is modeled. When—as is common in practice—the model's alphabet is taken to coincide with the unit inventory, this choice directly determines the events to which probability is assigned and, consequently, the quantities that enter the linking hypothesis relating surprisal to behavioral data.

Despite this importance, the choice of units has received comparatively little attention in surprisal theory. In practice, most studies inherit their units from the tokenizer of a pretrained language model or from the segmentation conventions of a particular eye-tracking corpus, treating the unit inventory as fixed rather than a variable to be controlled. This obscures a logically prior question: what *should* the units be? Answering it requires separating U, the countable (but not necessarily finite) unit inventory over which we wish to define surprisal, from Σ, the alphabet of the language model. We make this separation explicit below. Rather than advocating for a specific unit inventory, we develop a general formalism—in a slogan, "bring your own units"—that is compatible with any choice the modeler wishes to make, irrespective of the language model's native alphabet. To that end, we assume the modeler has a countable unit inventory U. This set may be finite, e.g., the set of phonemes in a language, or countably infinite, e.g., the set of orthographic words.[4]

A linguistic **utterance** is a string of units $\boldsymbol{u} \in U^*$. A **unit parser** is a stochastic map $\rho\colon \Sigma^* \rightsquigarrow U^*$ that maps each symbol string $\boldsymbol{\sigma} \in \Sigma^*$ to a probability distribution $\rho(\cdot \mid \boldsymbol{\sigma})$ over unit strings. In many languages, parsing into units is inherently ambiguous; we give a canonical example of such ambiguity from Mandarin Chinese.

Example 1. Let Σ be the set of Chinese characters, and let $U = \Sigma^+$. Consider the string $\boldsymbol{\sigma} =$ 乒乓球拍卖完了. The unit parser ρ ought to assign positive probability to (at least) two unit strings:

(1) a. $\rho(\boldsymbol{\sigma}) =$ 乒乓球 · 拍卖 · 完了
pīngpāngqiú pāimài wánle
ping pong ball auction finish-PERF
"The ping-pong ball auction is over."

b. $\rho(\boldsymbol{\sigma}) =$ 乒乓球拍 · 卖 · 完了
pīngpāngqiúpāi mài wánle
ping-pong-paddle sell finish-PERF
"The ping-pong paddles have been sold."

The character 拍 (*pāi*) is the pivot: it can end the compound 球拍 (*qiúpāi*, 'paddle') or begin 拍卖 (*pāimài*, 'auction'), yielding different unit strings from the same symbol string.

Our choice of a Chinese example is strategic. In many languages that use whitespace in their orthography, e.g., English and most other European languages, parsing into units is far more deterministic—for example, the Penn Treebank tokenization convention (Marcus et al., 1993) assigns a unique segmentation to every string. In what follows, we therefore make the simplifying assumption that ρ is *deterministic*: for every $\boldsymbol{\sigma} \in \Sigma^*$, there is exactly one unit string $\boldsymbol{u}$ with $\rho(\boldsymbol{u} \mid \boldsymbol{\sigma}) = 1$. Under this assumption, ρ reduces to a total function $\Sigma^* \to U^*$. In the remainder of the paper, we write $\rho(\boldsymbol{\sigma}) = \boldsymbol{u}$ and treat ρ as a function. We call the unit parser's inverse its **realization** $\rho^{-1} \subseteq U^* \times \Sigma^*$. In general, the realization is a relation, because one unit string may correspond to multiple symbol strings; this is true even in English.

Example 2. Let Σ be the set of cased ASCII characters, and let $U = \{\textsc{Hale}, \textsc{cited}, \textsc{Levy}\}$. Assuming ρ is deterministic, an English unit parser ought to map the following two symbol strings

(2) a. `Hale␣cited␣Levy.`
b. `Hale␣cited␣␣␣Levy.`

to the same unit string HALE·CITED·LEVY as they only differ in terms of whitespace. Thus, the realization ρ^{-1} is a non-functional relation.

3.1 Pushforwards

Suppose we have a language model p_Σ over the alphabet Σ, and we wish to define a language model p_U over the unit inventory U via the unit parser $\rho\colon \Sigma^* \to U^*$. The **pushforward** of p_Σ through ρ is given by the following expression

$$p_U(\boldsymbol{u}) \stackrel{\text{def}}{=} \sum_{\boldsymbol{\sigma} \in \rho^{-1}(\boldsymbol{u})} p_\Sigma(\boldsymbol{\sigma}), \tag{6}$$

where the sum ranges over all symbol strings $\boldsymbol{\sigma}$ that the unit parser maps to $\boldsymbol{u}$. For p_U to be a well-defined probability distribution, we require that ρ be a total function $\Sigma^* \to U^*$, i.e., every symbol string $\boldsymbol{\sigma} \in \Sigma^*$ is mapped to

[4] In addition to neologism, recursive morphological processes, e.g., compounding and derivation, produce unboundedly many distinct word forms, yielding a countably infinite set of orthographic words; see Pinker (1994, Ch. 5).

exactly one unit string. Equivalently, the fibers $\{\{\boldsymbol{\sigma} \mid (\boldsymbol{\sigma}, \boldsymbol{u}) \in \rho\} \mid \boldsymbol{u} \in U^*\}$ partition Σ^*, which is always true for the fibers of a total function. Without further assumptions on ρ, Eq. (6) is difficult to compute—the number of symbol strings related to $\boldsymbol{u}$ may be countably infinite.

3.2 Lost in Whitespace

Recent work (Oh and Schuler, 2024; Pimentel and Meister, 2024) proposes a formalism for computing the probability of the next unit in context given a symbol-level language model. We identify two conceptual issues with their shared approach—one mathematical and one linguistic.

Unit Inconsistency. Both papers formalize the conversion from a symbol-level language model to a unit-level model under three assumptions about the realization $\rho^{-1} \subseteq U^* \times \Sigma^*$: (i) ρ^{-1} is a monoid homomorphism,[5] i.e., ρ^{-1} is a function where $\rho^{-1}(u_1 \cdot \cdots \cdot u_T) = \rho^{-1}(u_1) \cdot \cdots \cdot \rho^{-1}(u_T)$; that is, each unit is realized in Σ independently, and the realization of a sequence of units is the concatenation of the individual realizations; (ii) Σ can be partitioned into two disjoint subsets Σ_1 and Σ_2, where Σ_1 contains symbols that mark a unit boundary and Σ_2 does not; (iii) each unit maps to a symbol string in $\Sigma_1 \circ \Sigma_2^*$, i.e., one boundary-marking symbol followed by zero or more continuation symbols. Assumption (i) implies that ρ^{-1} is a function. Assumption (ii) is motivated by the fact that many tokenizers prepend a whitespace character to the first token of each word, so that ␣ signals a unit boundary.[6] Consider the two symbol strings

(3) a. `Hale␣cited␣Levy.`
    b. `Levy␣cited␣Hale.`

which we would naturally expect to correspond to the following unit strings

(4) a. $\textsc{Hale}_1 \cdot \textsc{cited} \cdot \textsc{Levy}$
    b. $\textsc{Levy} \cdot \textsc{cited} \cdot \textsc{Hale}_2$

where $\textsc{Hale}_1 = \textsc{Hale}_2$. However, assumptions (i)–(iii) make such an equivalence impossible. In the first string, $\textsc{Hale}_1$ is string-initial so $\rho^{-1}(\textsc{Hale}_1) = \textsc{bos}\texttt{Hale}$, where BOS is the beginning-of-string symbol from Footnote 6. In the second, $\textsc{Hale}_2$ is preceded by ␣ so $\rho^{-1}(\textsc{Hale}_2) = \texttt{␣Hale}$. Because assumptions (i)–(iii) force ρ^{-1} to be a function, we have $\textsc{Hale}_1 \neq \textsc{Hale}_2$. Consequently, HALE is forced to be two distinct units depending on its context. Thus, the formalism cannot provide a coherent language model over units—the identity of a unit should not depend on whether it begins a string. Our framework sidesteps this problem because ρ^{-1} is a relation, not a function: the unit HALE can stand in the realization relation to *both* BOS`Hale` and ␣`Hale`, so a single unit suffices regardless of context.

Linguistic Adequacy. A second concern is linguistic. The approach of Oh and Schuler (2024) and Pimentel and Meister (2024) implicitly assumes that units can be recovered by grouping symbols in Σ according to the partition $\Sigma_1 \sqcup \Sigma_2$ from assumption (ii). But byte-pair encoding is a compression algorithm: the boundaries it induces are artifacts of corpus frequency, not of morphological or syntactic structure. Moreover, their framework requires every symbol to be classified as either a boundary marker (in Σ_1) or a continuation symbol (in Σ_2), uniformly across all contexts. Yet a comma is word-internal in `1,000` but marks a clause boundary in `end, he said`; an apostrophe is word-internal in `don't` but possessive-marking in `cat's`. No fixed partition can capture such distinctions in general, since the same character serves different roles in different environments. We remark that this concern is far from pedantic; Clark et al. (2025, §2.2.1) report discarding stimuli as the method does not properly handle punctuation.

3.3 Regular Unit Inventories

A technical challenge arises when the unit inventory is infinite, because language models operate over finite alphabets by definition, and the unit inventory U may be countably infinite, e.g., the set of all whitespace-delimited words. The key observation is that even an infinite U can be finitely represented whenever each unit is itself a string over some finite alphabet, i.e., $U \subseteq \Xi^*$, and U forms a *regular* subset of Ξ^*.[7] We call this the **regularity assumption**.

For many abstract linguistic units, the regularity assumption is well-motivated, e.g., it is widely es-

[5] The monoids in question are $(U^*, \cdot, \varepsilon)$ and $(\Sigma^*, \cdot, \varepsilon)$, i.e., the free monoids over U and Σ respectively, with concatenation as the binary operation and the empty string as the identity.

[6] A notable exception occurs at the beginning of a string: the first symbol belongs to Σ_2 regardless of the unit it represents. One can encode this by introducing an additional copy of all symbols in Σ_1 with a distinguished BOS symbol prepended.

[7] A language is *regular* if it is accepted by a finite automaton; see Pin (2025) for a comprehensive treatment. Also, note that regularity is a *sufficient*, but not necessary condition.

tablished that phonotactic constraints are regular (Kaplan and Kay, 1994; Heinz, 2018), as are many morphotactic rules (Koskenniemi, 1983; Beesley and Karttunen, 2003).[8] Because units at any of these levels are defined by regular constraints, regularity of U is a mild assumption in practice. Under this assumption, we can reduce operations on a potentially infinite U to operations over regular sets as follows. Let $\boldsymbol{\xi} \in \Xi^*$, and let $\textsc{sep} \notin \Xi$ be a distinguished separator symbol. We define

$$\begin{aligned} h \colon U &\to \Xi^* \textsc{sep} \\ u &\mapsto \boldsymbol{\xi} \textsc{sep}, \end{aligned} \tag{7}$$

which appends SEP to each unit's underlying string.[9] This extends to a monoid homomorphism $U^* \to (\Xi^* \textsc{sep})^*$ by defining $h(u_1 \cdot \cdots \cdot u_T) \stackrel{\text{def}}{=} h(u_1) \cdots \cdot h(u_T)$. Note that $h(U)$ is regular since U is regular by assumption and regular sets are closed under concatenation, and so $h(U^*) = h(U)^*$ is regular by standard closure properties.

3.4 Transduced Language Models

We now introduce a *computational* formalism for describing $\rho \colon \Sigma^* \to U^*$. First, let $\Delta \stackrel{\text{def}}{=} \Xi \sqcup \{\textsc{sep}\}$, following Eq. (7). Then, note that the composition $h \circ \rho \colon \Sigma^* \to (\Xi \sqcup \{\textsc{sep}\})^*$ takes Σ-strings and maps them to strings over the finite alphabet $\Xi \sqcup \{\textsc{sep}\}$. If we can compute $h \circ \rho$, we can apply h^{-1} to the output to map back to a unit string.

A **transducer** is a state machine encoding a string-to-string relation $f \subseteq \Sigma^* \times \Delta^*$. Formally, it is defined as a tuple $\mathsf{f} = (\mathcal{Q}, \Sigma, \Delta, \Omega, \mathcal{I}, \mathcal{F})$, where $\mathcal{Q}$ is a set of states,[10] Σ and Δ are the input and output alphabets, $\mathcal{I}, \mathcal{F} \subseteq \mathcal{Q}$ are the sets of initial and final states, and $\Omega \subseteq \mathcal{Q} \times (\Sigma \cup \{\varepsilon\}) \times (\Delta \cup \{\varepsilon\}) \times \mathcal{Q}$ is the set of transitions. For any two states $q, q' \in \mathcal{Q}$, we write $(q, \sigma, \delta, q') \in \Omega$ as shorthand for the transition $q \xrightarrow{\sigma:\delta} q'$. A transducer is called **finite** when $\mathcal{Q}$ is a finite set.[11] A function is called **rational** if it can be realized by a finite transducer.

[8] A notable exception is reduplication, which is not regular and thus falls outside the scope of standard finite-state methods. However, Dolatian and Heinz (2018) argue that 2-way finite transducers can nonetheless model reduplication.

[9] It will be important that h be **prefix free**, i.e., there do not exist units $u_1, u_2 \in U$ such that $h(u_1) \preceq h(u_2)$.

[10] In general, $\mathcal{Q}$ need not be finite and may encode auxiliary memory (e.g., a stack), modeling context-free behavior.

[11] Finite transducers are widely used for representing string-to-string functions in natural language processing (Roche and Schabes, 1997; Mohri, 1997), and their mathematical properties are well-understood; see Pin (2025, 2021) for standard operations such as composition and determinization. Here, we focus on rational functions, as they provide a compact framework for manipulating language models.

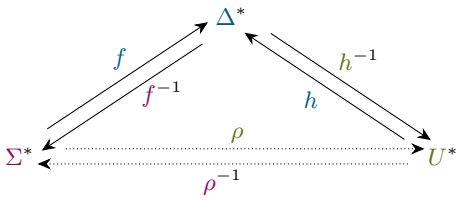

Figure 2: The unit parser $\rho \colon \Sigma^* \to U^*$ passes through Δ^*: the transducer f maps symbol strings to SEP-annotated strings, and h^{-1} splits on SEP and maps each segment to the unit it spells, recovering the unit string.

Define $f \stackrel{\text{def}}{=} h \circ \rho \colon \Sigma^* \to \Delta^*$. Note that f maps between two *finite* alphabets, Σ and Δ, even when the unit inventory U is infinite[12] because $U \subseteq \Xi^*$. If, in addition, f is rational,[13] Snæbjarnarson et al. (2026) provide a practical algorithm for computing the pushforward (see §3.1) under f, defined as

$$p_\Delta(\boldsymbol{\delta}) \stackrel{\text{def}}{=} \sum_{\boldsymbol{\sigma} \in f^{-1}(\boldsymbol{\delta})} p_\Sigma(\boldsymbol{\sigma}). \tag{8}$$

This gives us a distribution over Δ-strings; we now extract unit-level probabilities from it. Writing $\overrightarrow{p_U}$ for the prefix probability of p_U, the conditional next-unit probability is given by

$$\overrightarrow{p_U}(u \mid \boldsymbol{u}) = \overrightarrow{p_\Delta}(h(u) \mid h(\boldsymbol{u})). \tag{9}$$

The simplicity of Eq. (9) follows from the injectivity of h, by construction, and the fact that h is prefix free; see Footnote 9.

4 Regions of Interest

Experimenters often study predictions at spans that cover multiple units, such as sentences (Lau et al., 2017; Meister et al., 2021; Giulianelli et al., 2023), dialogue turns (Wallbridge et al., 2022, 2023), or discourse segments (Tsipidi et al., 2024, 2025), each of which spans multiple word-level units. To describe such spans formally, let $U^+ = U^* \setminus \{\varepsilon\}$. Given an utterance $\boldsymbol{u} = u_1 \cdots u_T \in U^+$ of T units, we write $\boldsymbol{u}_{[i,j)} = u_i \cdots u_{j-1}$ for $1 \leq i < j \leq T+1$ and refer to it as a **region of interest** (ROI; Giulianelli et al., 2024).

To predict reading time for an ROI $\boldsymbol{u}_{[i,j)}$ from a language model p_U over $\boldsymbol{u} \in U^*$, one must specify how to combine unit-level surprisals. The standard approach is to sum surprisals over the ROI (Smith

[12] There is a relevant line of work that extends classical finite automata and transducers with transitions labeled by predicates over potentially infinite alphabets, known as *symbolic* finite automata and transducers (van Noord and Gerdemann, 2001; Veanes et al., 2012; D'Antoni and Veanes, 2017).

[13] The regularity assumption is necessary for rationality of f whenever ρ surjects onto U^*.

and Levy, 2013; Nair and Resnik, 2023). Importantly, this sum yields the surprisal of the character sequence but omits the probability of SEP or EOS signaling the unit boundary, so it is not directly comparable with unit-level surprisal.

Computing ROI-level surprisal requires that their boundaries be compatible with finer-grained unit boundaries; otherwise, a unit may overlap two ROIs and cannot be unambiguously assigned to either. For example, consider the stimulus *Predictive power* with characters as units. A psycholinguist studying parafoveal preview (Rayner, 1975; Rayner et al., 1982; Blanchard et al., 1989) might define each ROI as the first three characters of a word. However, as shown in Ex. (5), the ROI P·R·E spans parts of two GPT-2 (Radford et al., 2019) tokens (P and redict), so the model's token-level probabilities alone cannot yield the surprisal of this character span. To resolve such mismatches, we can transform a language model from its native token alphabet to the character alphabet (Vieira et al., 2025).

(5) a. P·R·E·D·I·C·T·I·V·E·␣·P·O·W·E·R
Character units (ROIs underlined)

b. P redict ive ␣power
47 17407 425 1176
GPT-2 tokens

5 Surprisal Theory

Surprisal theory (Hale, 2001; Levy, 2008) posits that the incremental processing difficulty of language comprehension is a function of how unexpected an upcoming linguistic unit is given its context. The theory assumes an implicit human language model p_{H}, and predicts that the processing effort incurred by a unit is monotonically related to its surprisal. In practice, empirical tests of surprisal theory rely on large pretrained language models as proxies for p_{H} (Wilcox et al., 2023; Oh and Schuler, 2023; Shain et al., 2024; Kuribayashi et al., 2024). Additionally, evaluating the theory requires a **linking hypothesis**, i.e., a specification of how surprisal maps onto an observable dependent variable such as reading time. Much attention has been devoted to the functional form of this mapping, whether processing effort scales linearly (Smith and Levy, 2013; Shain et al., 2024; Wilcox et al., 2023), sublinearly (Brothers and Kuperberg, 2021), or superlinearly (Hoover et al., 2023) with surprisal, while Xu et al. (2023) find that the shape depends on the language model. However, as we argue in this paper, the choice of units and the aggregation strategy are equally consequential.

The training set $\{\boldsymbol{u}^n\}_{n=1}^N$ consists of N distinct utterances. We write T_n for the number of units in the n^{th} utterance. For each u_t^n given preceding context $\boldsymbol{u}_{<t}^n$, we measure a reading time $r_\pi(u_t^n, \boldsymbol{u}_{<t}^n)$ from one of the P participants. Because fixation durations are strictly positive and right-skewed, we model reading times with a log-normal generalized additive mixed model (GAMM; Wood, 2017). At position $t \in [T_n]$ of utterance n, let $\mathbf{x}_t^n = (x_{1,t}^n, \ldots, x_{J,t}^n)^\top$ denote the vector of J predictors. We model

$$\log r_\pi(u_t^n, \boldsymbol{u}_{<t}^n) = \mu_\pi(u_t^n, \boldsymbol{u}_{<t}^n) + \epsilon, \qquad (10)$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is Gaussian noise, σ^2 is the residual variance, and the log-mean is given by

$$\mu_\pi(u_t^n, \boldsymbol{u}_{<t}^n) = \sum_{j=1}^{J} f_j(x_{j,t}^n) + z_\pi(\mathbf{x}_t^n). \qquad (11)$$

Each f_j is a penalized smooth function of the j^{th} component $x_{j,t}^n$ of $\mathbf{x}_t^n$, so that the relationship between each predictor and reading time is learned nonparametrically. See §5.1 for a discussion of the predictors. The term $z_\pi(\mathbf{x}_t^n)$ captures participant-level random effects: a random intercept and by-participant random slopes for each predictor; see App. E for the full specification.

5.1 Baseline Predictors

A unit's **length** and **frequency** are standard baseline controls in eye-tracking regressions (Demberg and Keller, 2008; Smith and Levy, 2013; Goodkind and Bicknell, 2018): short, high-frequency units are more likely to be skipped, and when fixated, receive shorter fixation durations than longer, lower-frequency ones (Rayner and Raney, 1996; Kliegl et al., 2004). Accordingly, evaluations of contextual surprisal typically include both as baseline predictors (Wilcox et al., 2023; Opedal et al., 2024; Kuribayashi et al., 2024), with frequency operationalized as **unigram surprisal**.[14] Prior work either computes word frequencies on held-out data (Pimentel and Meister, 2024) or relies on precompiled lexical resources (e.g., Wilcox et al., 2023; Opedal et al., 2024; Re et al., 2025), using toolkits such as Speer (2022). However, these convenient methods introduce two nontrivial mismatches. First, Speer (2022) conflates distinct orthographic

[14] See Shain (2019, 2024) and Opedal et al. (2024) for discussion of unigram surprisal in analyzing reading times.

forms by stripping punctuation; for instance, it assigns AND the same probability as AND·,. Moreover, the resulting unigram distribution is not aligned with the language model used to derive contextual surprisal, complicating comparisons between frequency and contextual predictability. We thus follow Hopton et al. (2026) and estimate unigram surprisal directly from the language model: we sample text from the LM, process each sample through the transduced LM to obtain per-unit conditional probabilities, and average these over all positions. The resulting unigram distribution is consistent with the model's own distribution and is naturally defined for every unit in our inventories; see App. D for additional details. We include unigram surprisal estimated in this way as a baseline predictor in all analyses. Finally, to account for spillover effects we include the predictors for the preceding unit as controls (Rayner et al., 1983).

5.2 Predictive Power

We evaluate the contribution of surprisal by comparing two instances of the model in Eqs. (10) and (11): a **baseline model** $\widetilde{\varphi}$, in which the log-mean $\widetilde{\mu}_\pi(u_t^n, \boldsymbol{u}_{<t}^n)$ depends only on control predictors (unit length, unigram surprisal, and their spillover lags; see §5.1), and a **target model** φ that additionally includes contextual surprisal and its spillover lags (see App. E for the full specification). We fit both models on the N training utterances and evaluate them on a held-out test set of M utterances, where utterance m spans T_m positions. Treating the end-of-sequence symbol EOS as the $(T_m+1)^{\text{th}}$ unit, let $\mathcal{I} \stackrel{\text{def}}{=} \{(m,t) : m \in [M],\ 1 \le t \le T_m+1\}$ denote the set of held-out (utterance, position) pairs. For brevity, we write $r_t^m \stackrel{\text{def}}{=} r_\pi(u_t^m, \boldsymbol{u}_{<t}^m)$ and $\mu_t^m \stackrel{\text{def}}{=} \mu_\pi(u_t^m, \boldsymbol{u}_{<t}^m)$ (and $\widetilde{\mu}_t^m$ for the baseline). Writing the log-normal density from Eq. (10) as

$$\varphi\big(r_t^m \mid \mu_t^m, \sigma^2\big) \stackrel{\text{def}}{=} \frac{1}{r_t^m\,\sigma\sqrt{2\pi}} \exp\left(-\frac{(\log r_t^m - \mu_t^m)^2}{2\sigma^2}\right), \quad (12)$$

we measure the mean per-observation improvement in held-out log-likelihood, which is defined as

$$\Delta_{\text{llh}} \stackrel{\text{def}}{=} \frac{1}{|\mathcal{I}|} \sum_{(m,t)\in\mathcal{I}} \log \frac{\varphi(r_t^m \mid \mu_t^m, \sigma^2)}{\widetilde{\varphi}(r_t^m \mid \widetilde{\mu}_t^m, \widetilde{\sigma}^2)}, \quad (13)$$

where μ_t^m and $\widetilde{\mu}_t^m$ are the predicted log-means from the target and baseline models at position t of held-out utterance m, and σ and $\widetilde{\sigma}$ are the corresponding residual standard deviations, both estimated on the training set. A positive Δ_{llh} indicates that contextual surprisal captures variance in reading times beyond the baseline controls.

6 Experiments

We now evaluate the predictive power of surprisal theory under four unit inventories.

6.1 Unit Inventories

Tokens. The simplest option is to use the model's native tokens as units, i.e., $U = \Sigma$. This is a natural choice when the objective is to characterize the model itself, e.g., when comparing the impact of token granularity on surprisal (Oh and Schuler, 2025), or to evaluate the cognitive plausibility of token-like representations (Beinborn and Pinter, 2023; Nair and Resnik, 2023).[15] However, in other experimental paradigms, model tokens are a poor fit, as they rarely align with linguistically meaningful units (Church, 2020; Hofmann et al., 2021; Nair and Resnik, 2023). Another limitation of tokens is their coarse, model-dependent granularity, which can obscure effects that are naturally defined at finer spatial scales in the stimulus (Rayner, 1975; Schotter et al., 2012); see, for example, Ex. (5).

Characters. At the other extreme, individual characters can constitute units. Character-level surprisal may be useful when the ROIs are sublexical, such as punctuation (Rayner et al., 2000; Hill and Murray, 2000; Hirotani et al., 2006), morphological structure (Nair and Resnik, 2023), the first few characters to predict the skip rate of a word (Rayner et al., 1982; Blanchard et al., 1989), or as sublexical information used to improve surprisal estimates for larger units (Oh et al., 2021). We study character-level units in their own right, not merely as building blocks for computing ROI-level predictors.

Acontextual Words. A common choice is to define units using explicit orthographic rules. For instance, one could define a word-like notion by choosing the delimiter set Σ_1 from the partition $\Sigma_1 \sqcup \Sigma_2$ of Σ introduced in §3.2, such as whitespace (Wilcox et al., 2023; Pimentel and Meister, 2024), and splitting on those delimiters.[16] As illustrated in Figure 3, this kind of segmentation can be implemented with a finite transducer. However, this is

[15] BPE can be encoded as a finite state machine (Berglund and van der Merwe, 2023; Berglund et al., 2024).

[16] A design choice here is whether you also delete the delimiters or leave them in the resulting units.

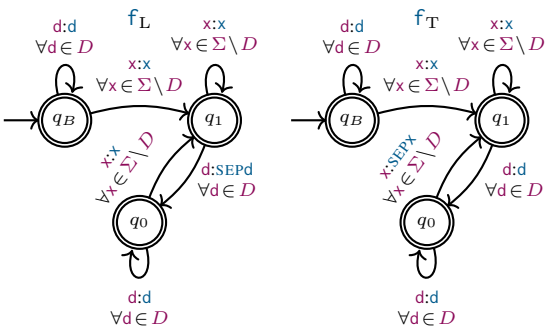

Figure 3: Two delimiter-insertion transducers. Left: f_L inserts SEP before the first delimiter following each unit. Right: f_T inserts SEP after that delimiter. These distinguish leading- and trailing-whitespace decoding (Oh and Schuler, 2024; Pimentel and Meister, 2024); see App. B.2 for discussion.

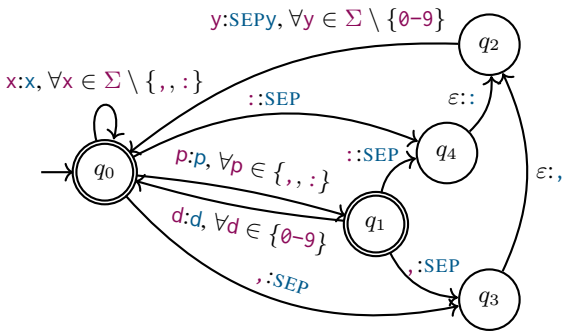

Figure 4: A rule from f_{ptb} showing contextual segmentation: a comma or colon is split off as its own unit (surrounded by SEPs) only when the *following* symbol is not a digit, e.g. `end,he` is split into three units, while `1,000` remains one. Adapted from Snæbjarnarson et al. (2026).

a modeling convenience inherited from how popular eye-tracking corpora such as Dundee (Kennedy et al., 2003), Provo (Luke and Christianson, 2018), or MECO (Siegelman et al., 2022) distribute their data, and is not meant to represent a linguistically adequate notion of a word. Its simplicity is also its main limitation: splitting on delimiters cannot express context-dependent boundaries. Consider, for example, how punctuation is typically considered its own unit in a context such as `long, tiring trip` but not in `1,000`. The same holds for internal apostrophes in contractions or in abbreviations. In fact, several recent studies have excluded all words attached to punctuation (Nair and Resnik, 2023; Gruteke Klein et al., 2024; Clark et al., 2025). However, research has long reported the systematic effects of punctuation on reading behavior (e.g., pauses and longer first pass times around commas) (Rayner et al., 2000; Hill and Murray, 2000; Hirotani et al., 2006). Handling such cases requires contextual segmentation rules.

Contextual Words. The acontextual assumption, i.e., that a fixed partition of Σ suffices to determine unit boundaries, is linguistically inadequate, because the same symbol can mark a boundary in one context but not another. Thus, instead of defining units via an explicit set of delimiters, it is common practice to use contextual segmentation rules, such as the Penn Treebank guidelines (Marcus et al., 1993), or Universal Dependencies (Nivre et al., 2017). Such units are linguistically informed, but introduce an additional challenge with token-level language models, where tokens are not compatible with the resulting units. Here we follow Snæbjarnarson et al. (2026) and encode each rule[17] as a finite transducer and then compose them left to right to obtain f_{ptb}; see App. B.3 for additional details. In contrast to the acontextual segmentation in Figure 3, this transducer determines word boundaries using contextual information. Consider, for example, the rule in Figure 4, which inserts a separator SEP before a comma (`,`) or a colon (`:`) only if the following symbol is not a digit.

[17]For an overview of the individual tokenization rules, see `TreebankWordTokenizer`.

6.2 (Re-)processing the MECO Corpus

To obtain fixation data for each unit inventory discussed in §6.1, we process the raw fixation data from the MECO dataset (Siegelman et al., 2022), which contains scanpaths from 46 readers recorded while reading 12 short excerpts drawn from Wikipedia articles. We use the English portion of the dataset. Following Re et al. (2025), we first obtain the unprocessed fixation data and use the predefined bounding boxes to match fixations with individual characters. We then tokenize the raw text and aggregate the fixation durations within the boundaries of each unit, to obtain three commonly used (Rayner, 1998) reading time measurements $r(u_t, \boldsymbol{u}_{<t})$: **first-fixation duration**, the duration of the first fixation on unit u_t; **gaze duration**, the sum of all first-pass fixations on u_t; and **total reading time**, the sum of all fixations on u_t, including any refixations. We exclude observations with zero reading time (i.e., unfixated units) and retain per-reader observations for use with mixed-effects models (App. E). For the character inventory, we additionally exclude observations whose surprisal is exactly zero, which arise at sub-token byte positions where BPE makes the next byte deterministic. In App. C.1, we visualize the resulting units and fixations; Table 4 reports the observation counts at each step of the pipeline.

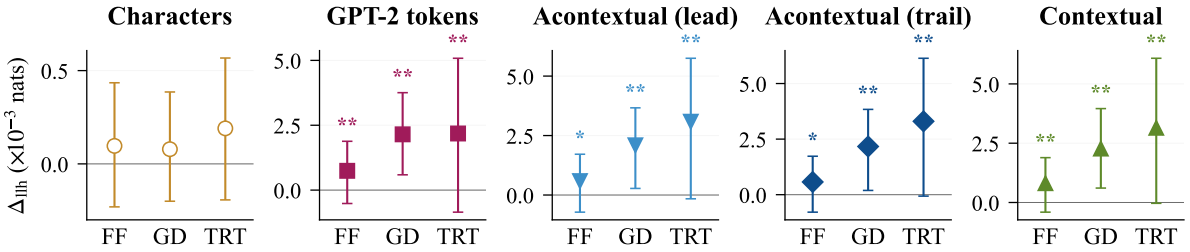

Figure 5: Per-observation Δ_{llh} ($\times 10^{-3}$ nats) for each unit inventory across reading-time measures (FF: first fixation, GD: gaze duration, TRT: total reading time). Points and whiskers show the mean and 95% trial-level bootstrap CI from leave-one-out cross-validation by trial. Significance is assessed via a paired permutation test (* $p < 0.05$; ** $p < 0.01$). Filled markers denote significant effects. Note that y-axis scales differ across panels: log-likelihoods are not comparable across inventories because the number and granularity of observations differ.

6.3 Estimating Surprisal

We use GPT-2 Small (Radford et al., 2019) as our symbol language model p_Σ, following previous work (Oh and Schuler, 2023). For the token inventory, surprisal is read directly from p_Σ. For all other inventories, we compose p_Σ with the appropriate finite transducer; see App. D for details.

6.4 Analysis

We fit the log-mean of Eq. (10) for both baseline and target models as a generalized additive mixed model (GAMM; Wood, 2017), which estimates the contribution of each predictor through a penalized smooth function. The residual standard deviations (σ and $\widetilde{\sigma}$) are estimated on the log scale of the training set. To assess generalization, we perform leave-one-out cross-validation by trial (12 folds): in each fold, we fit both models on 11 trials and compute Δ_{llh} (Eq. (13)) on the held-out trial. We report the per-observation Δ_{llh} with 95% confidence intervals obtained by trial-level bootstrap (1000 iterations). Significance of Δ_{llh} is assessed by a one-sided paired sign-flip permutation test over held-out log-likelihoods; see App. E for details.

6.5 Results

Figure 5 summarizes the results across four unit inventories and reading-time measures we consider; detailed per-measure results are given in Table 8. Adding surprisal as a predictor yields significant improvements in Δ_{llh} for all unit inventories, which is in line with previous findings (Goodkind and Bicknell, 2018; Wilcox et al., 2023), with the exception of the unit inventory of characters whose Δ_{llh} values are not significant ($p > 0.05$, paired permutation test) and markedly smaller, as single characters are rarely fixated individually. Among the acontextual words, the leading and trailing variants yield near-equal Δ_{llh}, with trailing slightly higher for gaze and total reading time and essentially unchanged for first-fixation duration.

A central lesson is that changing the unit of analysis changes the regression problem itself: different inventories induce different observations and controls (length, spillover, unigram surprisal), so absolute log-likelihoods and Δ_{llh} values are not directly comparable across inventories (Table 4). This lesson matters beyond the inventories considered here: existing work has evaluated surprisal at granularities from morphemes (Nair and Resnik, 2023) and phonemes (Brodbeck et al., 2022; Tezcan et al., 2023; Sohoglu et al., 2024) to sentences (Lau et al., 2017; Giulianelli et al., 2023) and discourse units (Tsipidi et al., 2024, 2025). More broadly, the framework applies wherever the unit of analysis must be specified—whether the dependent variable is reading time, neural signals (Frank et al., 2013, 2015; Kuribayashi et al., 2025), or a modified predictive distribution such as lossy-context surprisal (Futrell et al., 2020) or syntactic surprisal (Demberg and Keller, 2008; Arehalli et al., 2022).

7 Conclusion

We make a simple point: next-unit contextual surprisal is only well-defined relative to a choice of unit inventory. Yet existing work often inherits the language model's tokenizer. We present a formalism that makes this choice explicit and returns it to the modeler. We describe a principled way to derive unit-level surprisal from token-level language models. Because the choice of unit reshapes the regression problem and its baseline predictors, we argue it should be treated as a first-class modeling decision. We encourage future work to actively select the units most appropriate for their analysis.

Limitations

This study is primarily methodological, discussing the appropriate use of units and ROIs in surprisal theory and is therefore limited in scope.

Unit Inventories and ROIs. Our empirical evaluation is restricted to tokens, characters, and two families of word-like segmentations (contextual and acontextual). Beyond these units, researchers have studied several other unit inventories and ROIs, such as discourse units (e.g., clauses or elementary discourse units). Our framework naturally extends to such inventories, and we leave their empirical investigation to future work.

Language and Model Coverage. Our empirical results are restricted to English: what counts as a word varies with orthography and linguistic traditions, and many languages require different segmentation rules than those standard in English (Nivre et al., 2017). In addition, our analysis evaluates only GPT-2 Small on the MECO dataset, using GAMMs to predict reading times. Future studies could therefore broaden the empirical analyses to evaluate unit inventories and ROI choices across models and datasets.

Data Requirements. Our analysis is contingent on access to raw fixation data. Many published reading-time corpora distribute only pre-aggregated word-level reading times, and self-paced reading datasets are inherently bound to a fixed segmentation. In such cases, fixations cannot be re-aggregated to alternative unit boundaries, and our approach can only be applied if the chosen unit inventory is compatible with the corpus's pre-existing segmentation.

Computational Costs. Computing surprisal under the transduced language model requires marginalizing over all source strings that map to a given output, which incurs computational overhead that depends on the transducer and the source language model. In our experiments, we use the beam-search approximations described in Snæbjarnarson et al. (2026). We argue here that the resulting throughput (Table 6) is sufficient to make contextual surprisal estimation computationally feasible for typical psycholinguistic corpora. Estimating unigram surprisal is considerably more demanding and in practice requires parallelization across samples; see App. D for details.

Expressivity. Our framework inherits the expressivity limits of finite-state machinery: we assume the unit parser ρ is deterministic, and that it is rational, i.e., realizable by a finite transducer. Phenomena beyond this scope, such as genuinely ambiguous parsing and non-rational transformations such as those requiring context-free structure, fall outside the current framework and are left to future work.

Ethical Considerations

This work is a think piece about the role of units in psycholinguistic theory. The datasets we use are public and released with the consent of all participants. All personally identifiable information had been removed prior to our use of the data. As such, we do not see any ethical problems with this work.

Acknowledgments

The authors would like to thank Andreas Opedal, Francesco Ignazio Re, Jacob Hoover Vigly, Zach Hopton, Eleftheria Tsipidi and Mario Giulianelli for their valuable feedback and helpful discussions. VS is supported by the Pioneer Centre for AI, DNRF grant number P1. We used generative AI to assist with writing and with debugging code. The code and the writing were carefully reviewed and verified by the authors, who take full responsibility for the content of this paper.

References

Suhas Arehalli, Brian Dillon, and Tal Linzen. 2022. Syntactic surprisal from neural models predicts, but underestimates, human processing difficulty from syntactic ambiguities. In *Proceedings of the Conference on Computational Natural Language Learning*.

David A. Balota, Alexander Pollatsek, and Keith Rayner. 1985. The interaction of contextual constraints and parafoveal visual information in reading. *Cognitive Psychology*, 17(3).

Kenneth R. Beesley and Lauri Karttunen. 2003. *Finite State Morphology*.

Lisa Beinborn and Yuval Pinter. 2023. Analyzing cognitive plausibility of subword tokenization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Martin Berglund, Willeke Martens, and Brink van der Merwe. 2024. Constructing a BPE tokenization DFA. In *Implementation and Application of Automata*.

Martin Berglund and Brink van der Merwe. 2023. Formalizing BPE tokenization. In *Proceedings of the*

International Workshop on Non-Classical Models of Automata and Applications.

Harry E. Blanchard, Alexander Pollatsek, and Keith Rayner. 1989. The acquisition of parafoveal word information in reading. *Perception & Psychophysics*, 46(1).

Leonard Bloomfield. 1933. *Language*.

Christian Brodbeck, Shohini Bhattasali, Aura AL Cruz Heredia, Philip Resnik, Jonathan Z Simon, and Ellen Lau. 2022. Parallel processing in speech perception with local and global representations of linguistic context. *eLife*, 11.

Trevor Brothers and Gina R. Kuperberg. 2021. Word predictability effects are linear, not logarithmic: Implications for probabilistic models of sentence comprehension. *Journal of Memory and Language*, 116.

Kenneth Ward Church. 2020. Emerging trends: Subwords, seriously? *Natural Language Engineering*, 26(3).

Thomas Hikaru Clark, Moshe Poliak, Tamar Regev, A. J. Haskins, Caroline Robertson, and Edward Gibson. 2025. The relationship between surprisal and prosodic prominence in conversation reflects intelligibility-oriented pressures. *Cognitive Science*, 49(10).

Loris D'Antoni and Margus Veanes. 2017. The power of symbolic automata and transducers. In *Computer Aided Verification*.

Ferdinand de Saussure. 1997. *Deuxième cours de linguistique generale (1908-1909) : d'après les cahiers d'Albert Riedlinger et Charles Patois=Saussure's second course of lectures on general linguistics (1908-1909) : From the notebooks of Albert Riedlinger and Charles Patois*.

Vera Demberg and Frank Keller. 2008. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2).

R. M. W. Dixon and Alexandra Y. Aikhenvald. 2002. *Word: A Cross-Linguistic Typology*.

Hossep Dolatian and Jeffrey Heinz. 2018. Modeling reduplication with 2-way finite-state transducers. In *Proceedings of the Workshop on Computational Research in Phonetics, Phonology, and Morphology*.

Susan F. Ehrlich and Keith Rayner. 1981. Contextual effects on word perception and eye movements during reading. *Journal of Verbal Learning and Verbal Behavior*, 20(6).

Stefan L. Frank, Leun J. Otten, Giulia Galli, and Gabriella Vigliocco. 2013. Word surprisal predicts N400 amplitude during reading. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Stefan L. Frank, Leun J. Otten, Giulia Galli, and Gabriella Vigliocco. 2015. The ERP response to the amount of information conveyed by words in sentences. *Brain and Language*, 140.

Richard Futrell, Edward Gibson, and Roger P. Levy. 2020. Lossy-context surprisal: An information-theoretic model of memory effects in sentence processing. *Cognitive Science*, 44(3).

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2).

Mario Giulianelli, Luca Malagutti, Juan Luis Gastaldi, Brian DuSell, Tim Vieira, and Ryan Cotterell. 2024. On the proper treatment of tokenization in psycholinguistics. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Mario Giulianelli, Sarenne Wallbridge, and Raquel Fernández. 2023. Information value: Measuring utterance predictability as distance from plausible alternatives. In *The Conference on Empirical Methods in Natural Language Processing*.

Adam Goodkind and Klinton Bicknell. 2018. Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*.

Kyle Gorman. 2016. Pynini: A Python library for weighted finite-state grammar compilation. In *Proceedings of the SIGFSM Workshop on Statistical NLP and Weighted Automata*.

Keren Gruteke Klein, Yoav Meiri, Omer Shubi, and Yevgeni Berzak. 2024. The effect of surprisal on reading times in information seeking and repeated reading. In *Proceedings of the Conference on Computational Natural Language Learning*.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Martin Haspelmath. 2011. The indeterminacy of word segmentation and the nature of morphology and syntax. *Folia Linguistica*, 45(1).

Jeffrey Heinz. 2018. *The computational nature of phonological generalizations*.

Robin L. Hill and Wayne S. Murray. 2000. Commas and spaces: Effects of punctuation on eye movements and sentence parsing. In *Reading as a Perceptual Process*.

Masako Hirotani, Lyn Frazier, and Keith Rayner. 2006. Punctuation and intonation effects on clause and sentence wrap-up: Evidence from eye movements. *Journal of Memory and Language*, 54(3).

Charles F. Hockett. 1958. *A Course in Modern Linguistics*.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. Superbizarre is not superb: Derivational morphology improves BERT's interpretation of complex words. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Jacob Louis Hoover, Morgan Sonderegger, Steven T. Piantadosi, and Timothy J. O'Donnell. 2023. The plausibility of sampling as an algorithmic theory of sentence processing. *Open Mind*, 7.

Zachary William Hopton, Francesco Ignazio Re, Samuel Kiegeland, Andreas Opedal, Eleanor Chodroff, and Ryan Cotterell. 2026. Revisiting the estimation of unigram surprisal. Under review.

Ronald M. Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3).

Alan Kennedy, Robin Hill, and Joël Pynte. 2003. The Dundee corpus. In *Proceedings of the European Conference on Eye Movement*.

Reinhold Kliegl, Ellen Grabner, Martin Rolfs, and Ralf Engbert. 2004. Length, frequency, and predictability effects of words on eye movements in reading. *European Journal of Cognitive Psychology*, 16(1-2).

Kimmo Koskenniemi. 1983. *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production*. Ph.D. thesis, University of Helsinki.

Tatsuki Kuribayashi, Yohei Oseki, and Timothy Baldwin. 2024. Psychometric predictive power of large language models. In *Findings of the Association for Computational Linguistics: NAACL*.

Tatsuki Kuribayashi, Yohei Oseki, Souhaib Ben Taieb, Kentaro Inui, and Timothy Baldwin. 2025. Large language models are human-like internally. *Transactions of the Association for Computational Linguistics*, 13.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *SOSP*.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*, 41(5).

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3).

Alvin M. Liberman, Franklin S. Cooper, Donald P. Shankweiler, and Michael Studdert-Kennedy. 1967. Perception of the speech code. *Psychological Review*, 74(6).

Steven G. Luke and Kiel Christianson. 2018. The Provo corpus: A large eye-tracking corpus with predictability norms. *Behavior Research Methods*, 50.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*.

David Marr. 1982. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*.

Clara Meister, Tiago Pimentel, Patrick Haller, Lena Jäger, Ryan Cotterell, and Roger Levy. 2021. Revisiting the Uniform Information Density hypothesis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

George A. Miller and Kathryn Ojemann McKean. 1964. A chronometric study of some relations between sentences. *Quarterly Journal of Experimental Psychology*, 16(4).

Jeff Mitchell, Mirella Lapata, Vera Demberg, and Frank Keller. 2010. Syntactic and semantic factors in processing difficulty: An integrated measure. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Mehryar Mohri. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2).

Elliot Murphy. 2024. What is a word? *arXiv preprint arXiv:2402.12605*.

Sathvik Nair and Philip Resnik. 2023. Words, subwords, and morphemes: What really matters in the surprisal-reading time relationship? In *Findings of the Association for Computational Linguistics: EMNLP*.

Joakim Nivre, Daniel Zeman, Filip Ginter, and Francis Tyers. 2017. Universal Dependencies. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*.

Byung-Doh Oh, Christian Clark, and William Schuler. 2021. Surprisal estimators for human reading times need character models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Byung-Doh Oh and William Schuler. 2023. Why does surprisal from larger transformer-based language models provide a poorer fit to human reading times? *Transactions of the Association for Computational Linguistics*, 11.

Byung-Doh Oh and William Schuler. 2024. Leading whitespaces of language models' subword vocabulary pose a confound for calculating word probabilities. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Byung-Doh Oh and William Schuler. 2025. The impact of token granularity on the predictive power of language model surprisal. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Andreas Opedal, Eleanor Chodroff, Ryan Cotterell, and Ethan Wilcox. 2024. On the role of context in reading time prediction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Tiago Pimentel and Clara Meister. 2024. How to compute the probability of a word. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Jean-Éric Pin. 2021. *Handbook of Automata Theory*.

Jean-Éric Pin. 2025. *Mathematical Foundations of Automata Theory*.

Steven Pinker. 1994. *The Language Instinct*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8).

Keith Rayner. 1975. Parafoveal identification during a fixation in reading. *Acta Psychologica*, 39(4).

Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124(3).

Keith Rayner, Marcia Carlson, and Lyn Frazier. 1983. The interaction of syntax and semantics during sentence processing: eye movements in the analysis of semantically biased sentences. *Journal of Verbal Learning and Verbal Behavior*, 22(3).

Keith Rayner, Gretchen Kambe, and Susan A. Duffy. 2000. The effect of clause wrap-up on eye movements during reading. *The Quarterly Journal of Experimental Psychology Section A*, 53(4).

Keith Rayner and Gary E. Raney. 1996. Eye movement control in reading and visual search: Effects of word frequency. *Psychonomic Bulletin & Review*, 3(2).

Keith Rayner, Arnold D. Well, Alexander Pollatsek, and James H. Bertera. 1982. The availability of useful information to the right of fixation in reading. *Perception & Psychophysics*, 31(6).

Francesco Ignazio Re, Andreas Opedal, Glib Manaiev, Mario Giulianelli, and Ryan Cotterell. 2025. A spatio-temporal point process for fine-grained modeling of reading behavior. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30518–30538. Association for Computational Linguistics.

Michael Riley, Cyril Allauzen, and Martin Jansche. 2009. OpenFst: An open-source, weighted finite-state transducer library and its applications to speech and language. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Tutorial Abstracts*.

Emmanuel Roche and Yves Schabes. 1997. *Finite-State Language Processing*.

Elizabeth R. Schotter, Bernhard Angele, and Keith Rayner. 2012. Parafoveal processing in reading. *Attention, Perception, & Psychophysics*, 74(1).

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Cory Shain. 2019. A large-scale study of the effects of word frequency and predictability in naturalistic reading. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Cory Shain. 2024. Word frequency and predictability dissociate in naturalistic reading. *Open Mind*, 8.

Cory Shain, Clara Meister, Tiago Pimentel, Ryan Cotterell, and Roger Levy. 2024. Large-scale evidence for logarithmic effects of word predictability on reading time. *Proceedings of the National Academy of Sciences*, 121(10).

Noam Siegelman, Sascha Schroeder, Cengiz Acartürk, Hee-Don Ahn, Svetlana Alexeeva, Simona Amenta, Raymond Bertram, Rolando Bonandrini, Marc Brysbaert, Daria Chernova, et al. 2022. Expanding horizons of cross-linguistic research on reading: The Multilingual Eye-movement Corpus (MECO). *Behavior research methods*, 54(6).

Nathaniel J. Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3).

Vésteinn Snæbjarnarson, Samuel Kiegeland, Tianyu Liu, Reda Boumasmoud, Ryan Cotterell, and Tim Vieira. 2026. Transducing language models. In *The International Conference on Learning Representations*.

Ediz Sohoglu, Loes Beckers, and Matthew H. Davis. 2024. Convergent neural signatures of speech prediction error are a biological marker for spoken word recognition. *Nature Communications*, 15(1).

Robyn Speer. 2022. rspeer/wordfreq: v3.0.

Filiz Tezcan, Hugo Weissbart, and Andrea E Martin. 2023. A tradeoff between acoustic and linguistic feature encoding in spoken language comprehension. *eLife*, 12.

Eleftheria Tsipidi, Samuel Kiegeland, Franz Nowak, Tianyang Xu, Ethan Wilcox, Alex Warstadt, Ryan Cotterell, and Mario Giulianelli. 2025. The harmonic structure of information contours. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Eleftheria Tsipidi, Franz Nowak, Ryan Cotterell, Ethan Wilcox, Mario Giulianelli, and Alex Warstadt. 2024. Surprise! Uniform Information Density isn't the whole story: Predicting surprisal contours in long-form discourse. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Gertjan van Noord and Dale Gerdemann. 2001. Finite state transducers with predicates and identities. *Grammars*, 4(3).

Margus Veanes, Pieter Hooimeijer, Benjamin Livshits, David Molnar, and Nikolaj Bjorner. 2012. Symbolic finite state transducers: algorithms and applications. In *Proceedings of the Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*.

Tim Vieira, Benjamin Lebrun, Mario Giulianelli, Juan Luis Gastaldi, Brian Dusell, John Terilla, Timothy J. O'Donnell, and Ryan Cotterell. 2025. From language models over tokens to language models over characters. In *Proceedings of the International Conference on Machine Learning*.

Sarenne Wallbridge, Peter Bell, and Catherine Lai. 2023. Do dialogue representations align with perception? an empirical study. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*.

Sarenne Carrol Wallbridge, Catherine Lai, and Peter Bell. 2022. Investigating perception of spoken dialogue acceptability through surprisal. In *Annual Conference of the International Speech Communication Association, Interspeech*.

Ethan G. Wilcox, Tiago Pimentel, Clara Meister, Ryan Cotterell, and Roger P. Levy. 2023. Testing the predictions of surprisal theory in 11 languages. *Transactions of the Association for Computational Linguistics*, 11.

Ethan Gotlieb Wilcox, Jon Gauthier, Jennifer Hu, Peng Qian, and Roger Levy. 2020. On the predictive power of neural language models for human real-time comprehension behavior. In *Proceedings of the Cognitive Science Society*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Simon N. Wood. 2017. *Generalized Additive Models: An Introduction with R*, 2 edition.

Weijie Xu, Jason Chon, Tianran Liu, and Richard Futrell. 2023. The linearity of the effect of surprisal on reading times across languages. In *Findings of the Association for Computational Linguistics: EMNLP*.

Appendix Contents

A Notation Glossary

Symbol	Meaning
Alphabets and Strings	
Σ	Symbol alphabet (characters or tokens).
Δ	Output alphabet of the transducer.
Ξ	Finite alphabet over which units are strings; $U \subseteq \Xi^*$ and $\Delta = \Xi \sqcup \{\textsc{sep}\}$ (§3.3).
U	Unit inventory chosen by the modeler (countable, possibly infinite).
SEP	Distinguished separator symbol marking unit boundaries; $\textsc{sep} \notin \Xi$.
EOS	End-of-sequence symbol; $\textsc{eos} \notin \Sigma$.
Σ^*	Kleene closure: set of all finite strings over alphabet Σ, including ε.
Σ^+	Non-empty strings: $\Sigma^* \setminus \{\varepsilon\}$.
Units and Contexts	
T	Length of a string or utterance (number of symbols or units).
u_t	The t-th unit in an utterance; $u_t \in U$.
$\boldsymbol{u}$	Utterance: sequence of units, $\boldsymbol{u} = u_1 \cdots u_T \in U^*$.
$\boldsymbol{u}_{<t}$	Preceding-unit context.
$\boldsymbol{u}_{[i,j)}$	Region of interest (ROI): subspan $u_i \cdots u_{j-1}$.
Maps and Transducers	
ρ	Unit parser (stochastic map $\rho\colon \Sigma^* \rightsquigarrow U^*$); assumed deterministic ($\Sigma^* \to U^*$) in this paper.
ρ^{-1}	Realization: a relation $\rho^{-1} \subseteq U^* \times \Sigma^*$ mapping unit strings to symbol strings.
f	String-to-string relation $f \subseteq \Sigma^* \times \Delta^*$.
$\mathtt{f}$	Finite transducer with input alphabet Σ and output alphabet Δ.
D	Set of delimiter symbols; $D \subseteq \Sigma$.
$\mathtt{f}_{\mathrm{L}}$, $\mathtt{f}_{\mathrm{T}}$	Acontextual (delimiter-based) FSTs: leading and trailing SEP insertion.
$\mathtt{f}_{\mathrm{ptb}}$	Contextual FST implementing Penn Treebank segmentation rules.
h	Homomorphism $U \to \Xi^* \textsc{sep}$ appending SEP to each unit's underlying string; extended to U^* by concatenation (Eq. (7)).
h^{-1}	Inverse of h: splits on SEP and maps each segment to its unit (§3.4).
Probability and Surprisal	
p_Σ	Source/token-level language model (probability distribution) over Σ^*.
p_Δ	Transduced language model over Δ^*, defined as $p_\Sigma \circ \mathtt{f}$.
p_U	Unit-level language model over U^*.
p_{H}	Implicit human language model assumed by surprisal theory.
s	Surprisal of a unit in context: $-\log \overrightarrow{p_U}(u_t \mid \boldsymbol{u}_{<t})$.
Reading-time Analysis	
$r_\pi(u_t^n, \boldsymbol{u}_{<t}^n)$	Reading-time measurement (first-fixation, gaze, or total) for unit u_t^n from participant π.
$\mathbf{x}_t^n$	Predictor vector $(x_{1,t}^n, \ldots, x_{J,t}^n)^\top$ at position t of utterance n.
N, n	Number of training utterances; index over training utterances.
M, m	Number of held-out (test) utterances; index over test utterances.
$\mathrm{LL}_{\mathrm{bl}}$, $\mathrm{LL}_{\mathrm{tgt}}$	Mean per-observation held-out log-likelihood of baseline / target GAMM.
Δ_{llh}	Improvement in held-out log-likelihood: $\mathrm{LL}_{\mathrm{tgt}} - \mathrm{LL}_{\mathrm{bl}}$.

Table 1: Notation used throughout the paper.

B Transducers

Here we provide additional details on the FSTs used in §6. We implement all FSTs in Pynini (Gorman, 2016), a Python library for compiling and composing finite transducers that builds on OpenFST (Riley et al., 2009).

B.1 Characters

As shown by Snæbjarnarson et al. (2026) (see Figure 6), the transformation from tokens to characters can be encoded using an FST. In this work, we do not compile the FST for the experiments in App. D; instead, we use the algorithms and implementations of Vieira et al. (2025) to transform p_Σ into a character-level model.

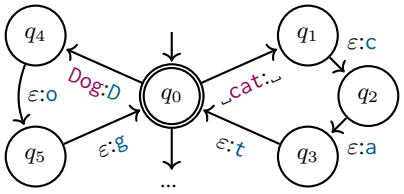

Figure 6: A finite transducer for mapping a token-level LM to characters, illustrated with paths for ␣cat and Dog. Adapted from Snæbjarnarson et al. (2026).

B.2 Acontextual Words

Recent work argues that the common leading-whitespace convention of many BPE tokenizers—where the space preceding a word is bundled with the word's first token—induces a misallocation of surprisal, and that probability mass should instead be attributed as trailing whitespace, i.e., assigned to the preceding word (Oh and Schuler, 2024; Pimentel and Meister, 2024). At the same time, Giulianelli et al. (2024) argue that there is no universally correct convention and the attribution of whitespace should be chosen to match the experimental setup. Note that under the definition of Oh and Schuler (2024) and Pimentel and Meister (2024), leading-versus-trailing whitespace decoding can be interpreted as an aggregation method that specifies how probability mass is redistributed across unit boundaries. Here, we express leading and trailing attribution as two finite transducers, $\mathtt{f}_\mathrm{L}$ and $\mathtt{f}_\mathrm{T}$; see Figure 3. In our experiments, we set the delimiter set to $D = \{␣\}$, so whitespace is the sole signal of a unit boundary.

B.3 Contextual Words

We represent contextual words using the rules described in the Penn Treebank annotation guidelines and encode each rule[18] as a small context-dependent string-rewrite transducer (see Figure 4 for an example of such a rule), using Pynini's rewrite calculus (e.g., replace operations with explicit left/right contexts and boundary conditions). The full tokenizer is obtained by composing these rule transducers left-to-right, yielding a single transducer that maps input strings to their PTB-style tokenized form.

B.4 Transducer Sizes

Table 2 reports the number of states and arcs of each FST $\mathtt{f}$ used in the experiments, together with its number of *universal* states, i.e., those states, where the corresponding input-projected FSA accepts every symbol $\sigma \in \Sigma$ (see Snæbjarnarson et al. (2026) for a detailed discussion). Universal states can be handled more efficiently in the algorithms provided by Snæbjarnarson et al. (2026), so transducers with a larger universal fraction yield a higher throughput (Syms/s). Table 3 complements this by reporting, for each pair of inventories, the percentage of units in the row inventory whose character span is also a unit in the column inventory (ignoring whitespace attribution).

Note that the two delimiter-insertion transducers $\mathtt{f}_\mathrm{L}$ and $\mathtt{f}_\mathrm{T}$ exhibit a size asymmetry despite being drawn as equivalent three-state machines in Figure 3. This is because the compiled FSTs allow only one

[18] `TreebankWordTokenizer`.

Transducer	States	Arcs	Universal
Acontextual (leading)	3	517	3 (all)
Acontextual (trailing)	258	1,029	258 (all)
Contextual	361	35,210	78 (partial)

Table 2: Size of each finite transducer f used in the experiments, together with the number of universal states (Snæbjarnarson et al., 2026); "all" indicates that every state is universal.

	GPT-2 token	Acontextual	Contextual
GPT-2 tokens	—	71.4%	83.6%
Acontextual	84.5%	—	92.4%
Contextual	91.5%	85.5%	—

Table 3: Pairwise unit overlap: percentage of units in each row inventory whose content span is also a unit in the column inventory.

output symbol per arc, while the figure uses a shorthand to draw arcs with multiple-symbol outputs. Each such arc is compiled into a chain of two arcs through an auxiliary state, with one auxiliary per input-byte value. In $\mathtt{f}_{\mathrm{L}}$ the multi-symbol output sits on the delimiter arc $q_1 \xrightarrow{\mathtt{d}:\mathrm{SEP}\mathtt{d}} q_0$, so only $|D|$ auxiliary states are introduced; with $D = \{\mathtt{␣}\}$ the single resulting auxiliary is merged away by minimization, leaving three states. In $\mathtt{f}_{\mathrm{T}}$ the multi-symbol output sits on the arc $q_0 \xrightarrow{\mathtt{x}:\mathrm{SEP}\mathtt{x}} q_1$, so $|\Sigma \setminus D|$ auxiliaries are introduced; with a byte alphabet ($|\Sigma| = 256$) that is 255 auxiliaries whose distinct output labels prevent merging, giving $3 + 255 = 258$ states.

B.5 Unit Overlap

GPT-2 tokens and acontextual words share 71.4% of their units; the 708 GPT-2 units absent from the acontextual inventory are predominantly punctuation marks that BPE splits off (126 commas, 96 periods) and BPE subword fragments (e.g., Jan, us, bee from words like *Janus*, *beekeeper*). Conversely, the 325 acontextual units absent from GPT-2 are words that BPE splits into multiple tokens (e.g., *Janus*, *thylacine*, *performance-enhancing*). Acontextual and contextual words overlap at 92.4%; the main source of disagreement is punctuation: the contextual inventory splits off 129 commas and 14 periods that the acontextual inventory attaches to the preceding word (e.g., acontextual ORGANIZATIONS, vs. contextual ORGANIZATIONS | ,). Possessives ('S) and quotation marks (“, ”) account for the remaining contextual-only units.

C Dataset Details

We reprocess the English portion of the MECO eye-tracking corpus (Siegelman et al., 2022), which contains scanpaths from 46 readers recorded while reading 12 short Wikipedia excerpts. Table 4 reports the number of observations at each stage of the analysis pipeline, from raw units to the final GAMM input, for each of the unit inventories.

Inventory	n_{units}	n_{obs}	n_{lag}
GPT-2 tokens	2,478	35,352	34,388
Acontextual (leading)	2,095	34,265	33,301
Acontextual (trailing)	2,095	34,265	33,301
Contextual	2,264	34,478	33,514
Characters	13,226	44,325	43,361

Table 4: Pipeline observation counts per unit inventory. n_{units}: total units across all 12 trials. n_{obs}: per-reader observations after excluding zero reading times (unfixated units). n_{lag}: after dropping the first two units of each (reader, trial) pair, which lack values for the spillover lags. See Table 7 for the final counts entering the GAMM.

C.1 Unit and Fixation Visualizations

Figures 7–10 show the same trial (Reader 3, Text 1 from the MECO English corpus) segmented under the unit inventories used in our experiments, together with the recorded fixation data. The leading and trailing acontextual inventories segment the text into identical content spans (differing only in whitespace attribution) and are represented by a single figure.

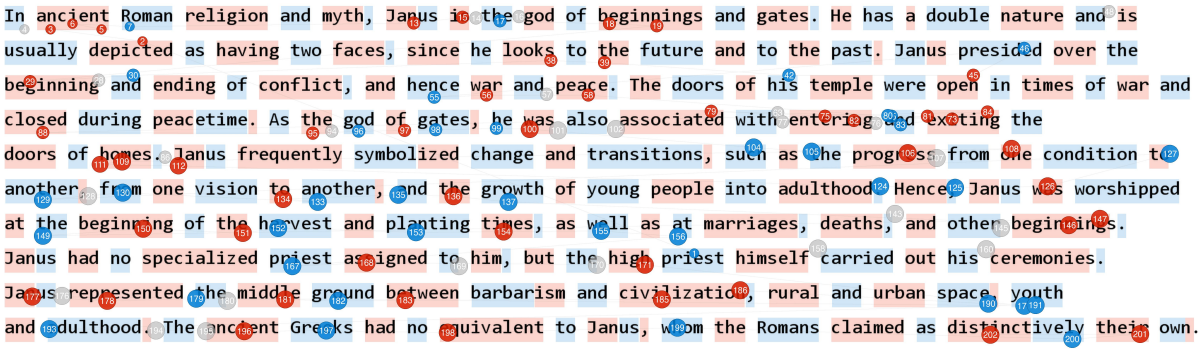

Figure 7: Units and fixations for **model tokens** (BPE, leading delimiter). Reader 3, Text 1 (MECO English).

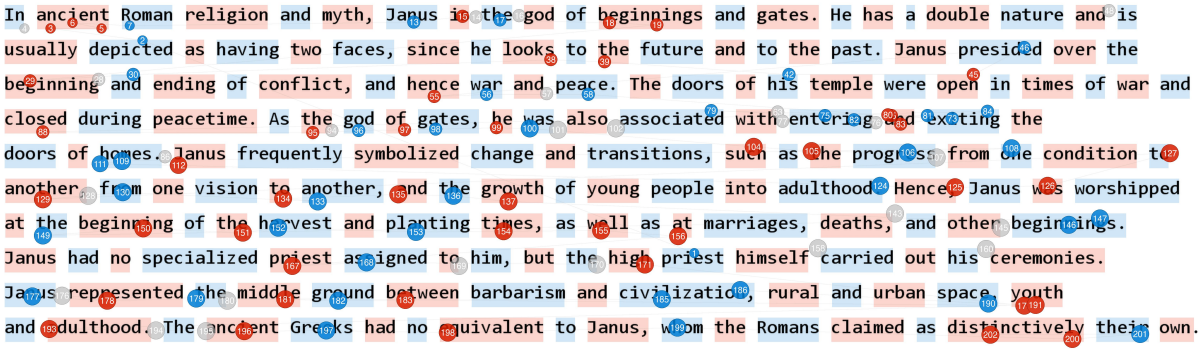

Figure 8: Units and fixations for **acontextual words**. Reader 3, Text 1 (MECO English).

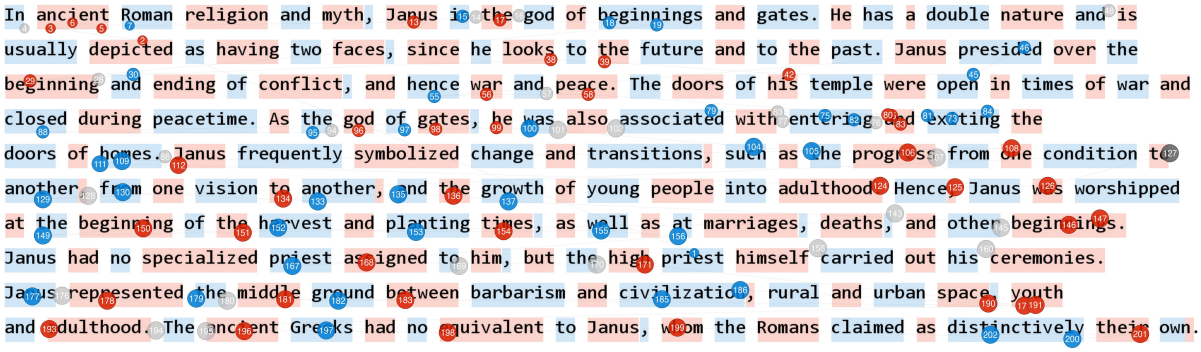

Figure 9: Units and fixations for **contextual words**. Reader 3, Text 1 (MECO English).

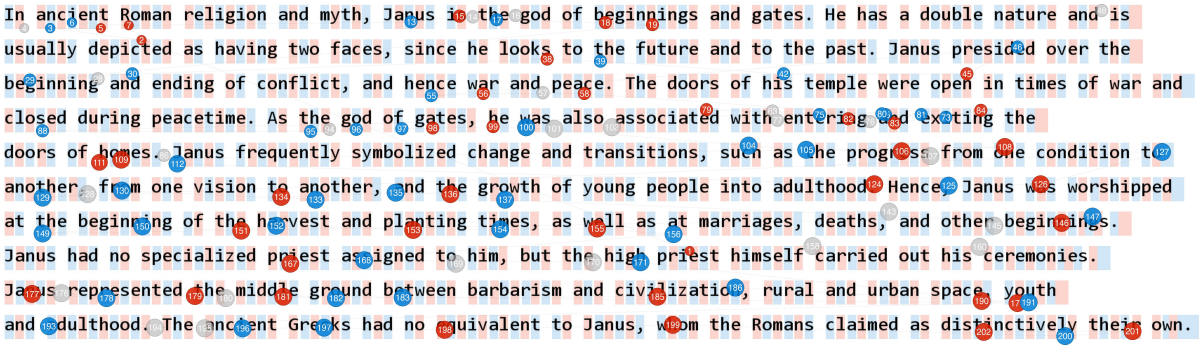

Figure 10: Units and fixations for **character-level units** (leading delimiter). Reader 3, Text 1 (MECO English).

D Experimental Details

To compute surprisal estimates for the experiments in §6, we use the implementation by Snæbjarnarson et al. (2026) to compose GPT-2 Small[19] (Radford et al., 2019) from the 🤗 Hugging Face hub (Wolf et al., 2020) with the respective transducers described in §3. To convert token-level models to character-level, we use GenLM.bytes.[20] To quickly compute next-token/byte distributions, we use vLLM (Kwon et al., 2023).

D.1 Computing Surprisal

Both contextual surprisal and unigram surprisal are computed under the transduced language model $p_\Delta = p_\Sigma \circ \mathtt{f}$, from the next-unit conditional distribution $\overrightarrow{p_U}(u \mid \boldsymbol{u}) = \overrightarrow{p_\Delta}(h(u) \mid h(\boldsymbol{u}))$ of Eq. (9). Following h as defined in Eq. (7), each unit's byte extension ends with a SEP that marks its right boundary. Per-unit conditional probability is therefore the ratio

$$\overrightarrow{p_U}(u \mid \boldsymbol{u}_{<t}) = \frac{\overrightarrow{p_\Delta}\big(h(\boldsymbol{u}_{<t}) \cdot h(u)\big)}{\overrightarrow{p_\Delta}\big(h(\boldsymbol{u}_{<t})\big)}, \tag{14}$$

in which the numerator is a byte-level prefix mass closed off by the SEP carried in $h(u)$, making h prefix-free (see footnote 9). The two quantities (contextual vs. unigram surprisal) differ only in how they consume this conditional: contextual surprisal scores the unit that actually occurred at each position, whereas unigram surprisal computes the marginal $\overrightarrow{p_U}(u) = \mathbb{E}_{\boldsymbol{u}_{<t}}[\overrightarrow{p_U}(u \mid \boldsymbol{u}_{<t})]$ with respect to contexts $\boldsymbol{u}_{<t}$ sampled from the LM, with u held fixed. Both share the hyperparameters listed in Table 5.

Contextual surprisal. For each of the 12 MECO trials we first apply the transducer $\mathtt{f}$ at the trial level to obtain the transduced string $\boldsymbol{\delta} \in \Delta^*$, then score its symbols left-to-right. Each step issues one call to the fast next-symbol decomposition (`decompose_next`) algorithm of Snæbjarnarson et al. (2026, §C.4), which returns the full next-symbol distribution over Δ. Consecutive calls are cached, so advancing the context from $\boldsymbol{\delta}_{<t}$ to $\boldsymbol{\delta}_{<t+1}$ extends the cached decomposition by a single symbol rather than recomputing from scratch; low-probability beams are pruned during expansion using the thresholds in Table 5.

Unigram surprisal. Unigram surprisal can be estimated using the same LM that estimates the surprisal of u under p_U by marginalizing over contexts (Hopton et al., 2026). Because the full marginalization is intractable, we compute a Monte Carlo estimate by drawing S samples from the LM, and for each sample, average the next-unit conditional $\overrightarrow{p_U}(u \mid \boldsymbol{u}_{<t})$ over every unit position t. For the **GPT-2 token** and **character** inventories, the unit alphabet coincides with the native output alphabet of the LM, so the conditional $\overrightarrow{p_U}(u \mid \boldsymbol{u}_{<t})$ is read off directly from the LM. For **acontextual** and **contextual** inventories, units are defined through $\mathtt{f}$ and $\overrightarrow{p_U}(u \mid \boldsymbol{u}_{<t})$ must be recovered from $\overrightarrow{p_\Delta}$: We first transduce each sample from p_Σ to its target string $\boldsymbol{\delta}^s \in \Delta^*$ and locate the SEP positions $b_1 < \cdots < b_{K_s}$ that mark unit boundaries. At each boundary b_k, we cache the closed prefix mass $z = \log \overrightarrow{p_\Delta}(\boldsymbol{\delta}^s_{\leq b_k})$, i.e., the prefix mass through and including the SEP at position b_k, and score every candidate unit $u \in U$ by evaluating $z_u = \log \overrightarrow{p_\Delta}(\boldsymbol{\delta}^s_{\leq b_k} \cdot h(u))$, so that the next-unit conditional is $\overrightarrow{p_U}(u \mid \boldsymbol{u}_{<t}) = \exp(z_u - z)$ exactly, following Eq. (9).

Efficient scoring for the transduced LM. Rather than computing the full next-distribution (`decompose_next`), we use the single-symbol scoring routine of Snæbjarnarson et al. (2026) (introduced there for cross-entropy evaluation), which decomposes only $\boldsymbol{\delta} \cdot \delta$ for a specified target symbol and returns $\log \overrightarrow{p_\Delta}(\boldsymbol{\delta} \cdot \delta)$ directly, skipping the full-vocabulary expansion and the final normalization. We apply this routine one target symbol at a time along the byte extension of each unit, accumulating log prefix masses and recovering the unit's conditional prefix probability by subtracting the cached boundary probability mass. Since all $|U|$ unit extensions at a given boundary begin from the same prefix, their decomposition is shared across units; units with common byte prefixes additionally reuse cached partial extensions.

[19] openai-community/gpt2
[20] genlm-bytes

Parameters. Table 5 lists the hyperparameters for both computations. The beam-search and transduced-LM parameters are shared; the sampling block applies only to unigram estimation.

Parameter	Value	Role
GenLM.bytes (Vieira et al., 2025)		
Beam size (K)	5	Maximum beam width; keeps the K highest-probability beams.
Beam prune threshold	0.001	Drop beams whose probability mass falls below this threshold.
Transduced-LM inference (Snæbjarnarson et al., 2026)		
f prune threshold (τ)	0.005	Drop FST paths with probability mass below τ.
Max expand steps	5	Halt expansion of non-universal states after this many steps.
Expand stop mass	0.01	Halt expansion once the relative remaining probability mass falls below this value.
Sampling for estimating unigram surprisal		
LM p_Σ	GPT-2 Small	Source language model used in all experiments.
# samples S	500	Number of samples drawn from p_Σ in the unigram estimator.
Max length	50	Maximum number of tokens per sample.
Batch size	64	Batch size used when sampling from p_Σ.

Table 5: Hyperparameters used to estimate contextual and unigram surprisal. The beam-search parameters apply to both estimators; the sampling parameters apply only to the unigram estimator.

D.2 GPU Usage & Runtime

All experiments were run on NVIDIA GeForce RTX 4090 GPUs and RTX 3090 GPUs, each with 24 GB of GPU memory. Table 6 reports scoring throughput for contextual surprisal. The acontextual FSTs process approximately 200 target symbols per second; at this rate, scoring the 12 MECO English trials takes approximately one minute on a single GPU. The contextual FST is more than an order of magnitude slower, due to its large number of states and arcs (see Table 2) and the fact that many of its states are not universal, requiring additional computation to traverse the FST until hitting a universal state; see App. B.4 for a brief discussion and Snæbjarnarson et al. (2026) for a detailed discussion on universality.

Table 6: Contextual surprisal scoring throughput per transducer on GPT-2 Small, aggregated over the 12 MECO English trials. "Symbols" is the total number of Δ-symbols scored across all trials; "Syms/s" is the corresponding throughput. The character-level model is omitted because character surprisal is obtained via Vieira et al. (2025) rather than through an FST composition.

Transducer	Symbols	Time (s)	Syms/s
Acontextual (leading)	15,309	72.0	212.8
Acontextual (trailing)	15,309	75.2	203.6
Contextual	13,407	1117.7	12.0

Unigram estimation is considerably more expensive than contextual surprisal, because every SEP-boundary of every sample requires an extension of the cached prefix mass for every unit in U. On GPT-2, typical per-sample scoring of a 211-byte sample takes on the order of 20 seconds for acontextual (leading), 60 seconds for acontextual (trailing), and 8 minutes for the contextual transducer; the relative ordering mirrors Table 6 because the same FST-decomposition step is the bottleneck in both pipelines. Run sequentially, at 500 samples per unit inventory, this translates to roughly 3 hours (acontextual leading), 8 hours (acontextual trailing), and 2–3 days (contextual) of sequential single-GPU compute, so unigram estimation in practice requires parallelization. Since the outer sample loop runs independently across s, we chunk the 500-sample runs into independent jobs.

E GAMM Specification

We model log reading time as a generalized additive mixed model (Wood, 2017) fitted with `bam()` from `mgcv` in R, using fast restricted maximum likelihood (`method="fREML"`, `discrete=TRUE`). Each continuous predictor enters the log-mean as a cubic regression spline with up to six basis functions (`bs="cr"`, `k=6`); the model further includes a random intercept for participant and by-participant random slopes for every continuous predictor.

Baseline model ($\widetilde{\varphi}$).

```
log(rt) ~ s(length, bs="cr", k=6)
  + s(length_prev, bs="cr", k=6)
  + s(length_prev2, bs="cr", k=6)
  + s(unigram_surprisal, bs="cr", k=6)
  + s(unigram_surprisal_prev, bs="cr", k=6)
  + s(unigram_surprisal_prev2, bs="cr", k=6)
  + s(length, participant, bs="re")
  + s(length_prev, participant, bs="re")
  + s(length_prev2, participant, bs="re")
  + s(unigram_surprisal, participant,
      bs="re")
  + s(unigram_surprisal_prev, participant,
      bs="re")
  + s(unigram_surprisal_prev2, participant,
      bs="re")
  + s(participant, bs="re")
```

Target model (φ). The target model adds contextual surprisal and its two spillover lags to the baseline:

```
log(rt) ~ ... [baseline terms] ...
  + s(surprisal, bs="cr", k=6)
  + s(surprisal_prev, bs="cr", k=6)
  + s(surprisal_prev2, bs="cr", k=6)
  + s(surprisal, participant, bs="re")
  + s(surprisal_prev, participant, bs="re")
  + s(surprisal_prev2, participant, bs="re")
```

Here `s(x, bs="cr", k=6)` denotes a cubic regression spline with 6 basis functions; `length` is unit length in characters; `unigram_surprisal` is unigram surprisal (see §5.1); `surprisal` is contextual surprisal from the language model; `_prev` and `_prev2` denote spillover from the first and second preceding unit, respectively. Each predictor enters as both a population-level smooth and a by-participant random slope `s(x, participant, bs="re")`; a random intercept for participants is included via `s(participant, bs="re")`. For the character inventory, unit length is constant (every unit is a single character), so the `length`, `length_prev`, and `length_prev2` smooths are omitted from both the baseline and target formulas; the remaining terms and the random-effects structure are unchanged.

Paired permutation test. Significance of Δ_{llh} is assessed by a one-sided paired permutation test on the per-observation held-out log-likelihood differences between the target and baseline models, with $B = 1000$ sign-flip permutations.

F Additional Results

Table 8 reports detailed GAMM results for each reading-time measure: mean per-observation held-out log-likelihood for the baseline (LL_{bl}) and target (LL_{tgt}) models, along with the improvement Δ_{llh} and 95% trial-level bootstrap CIs. Note that absolute log-likelihoods are not comparable across unit inventories because the number and granularity of observations differ. Table 7 reports the number of observations entering the regression for each inventory, after the additional exclusions specific to the GAMM input.

Inventory	n_{GAMM}
GPT-2 tokens	34,388
Acontextual (leading)	33,301
Acontextual (trailing)	33,301
Contextual	33,472
Characters	38,581

Table 7: Final number of observations entering the GAMM per unit inventory, starting from n_{lag} (Table 4) after additionally excluding observations with missing unigram surprisal (Contextual only: 42 obs for “ and ”) or zero surprisal (Characters only: 4,780 sub-token byte positions where the byte distribution is deterministic under BPE).

Table 8: GAMM results across reading-time measures. LL_{bl}/LL_{tgt}: mean per-obs. held-out log-likelihood of the baseline/target; Δ_{llh}: improvement ($\times 10^{-3}$ nats) with 95% trial-level bootstrap CI in brackets. Significance via paired permutation test (see App. E): $^{*}\ p < 0.05$; $^{**}\ p < 0.01$.

	First fixation				*Gaze duration*				*Total reading time*			
Inventory	LL_{bl}	LL_{tgt}	Δ_{llh}	p	LL_{bl}	LL_{tgt}	Δ_{llh}	p	LL_{bl}	LL_{tgt}	Δ_{llh}	p
Characters	−0.4999	−0.4998	$0.10_{[-0.23,\ 0.43]}$	0.227	−0.5063	−0.5062	$0.08_{[-0.20,\ 0.39]}$	0.260	−0.5651	−0.5649	$0.19_{[-0.19,\ 0.57]}$	0.098
GPT-2 tokens	−0.4767	−0.4760	$0.74^{**}_{[-0.52,\ 1.88]}$	0.007	−0.5875	−0.5854	$2.15^{**}_{[0.59,\ 3.76]}$	<0.001	−0.7296	−0.7274	$2.18^{**}_{[-0.85,\ 5.08]}$	<0.001
Acontextual (leading)	−0.4737	−0.4731	$0.58^{*}_{[-0.72,\ 1.71]}$	0.026	−0.6020	−0.6000	$2.09^{**}_{[0.28,\ 3.67]}$	<0.001	−0.7450	−0.7419	$3.08^{**}_{[-0.16,\ 5.75]}$	<0.001
Acontextual (trailing)	−0.4736	−0.4731	$0.57^{*}_{[-0.79,\ 1.73]}$	0.030	−0.6018	−0.5996	$2.17^{**}_{[0.19,\ 3.84]}$	<0.001	−0.7447	−0.7414	$3.30^{**}_{[-0.06,\ 6.14]}$	<0.001
Contextual	−0.4747	−0.4739	$0.81^{**}_{[-0.41,\ 1.89]}$	0.003	−0.5988	−0.5965	$2.27^{**}_{[0.61,\ 3.96]}$	<0.001	−0.7413	−0.7381	$3.15^{**}_{[-0.03,\ 6.08]}$	<0.001

Figure 11 shows the approximate F-statistics for all fixed-effect smooth terms in the full-data GAMM fit, broken down by predictor group and reading-time measure. We observe two patterns: First, the current-unit surprisal smooth is significant for every unit inventory and every reading-time measure. Second, length and spillover predictors contribute primarily to the word-like inventories (tokens, acontextual, contextual) and to the later measures (gaze duration and total reading time); at the character level, the length smooth is absent because all units share the same length.

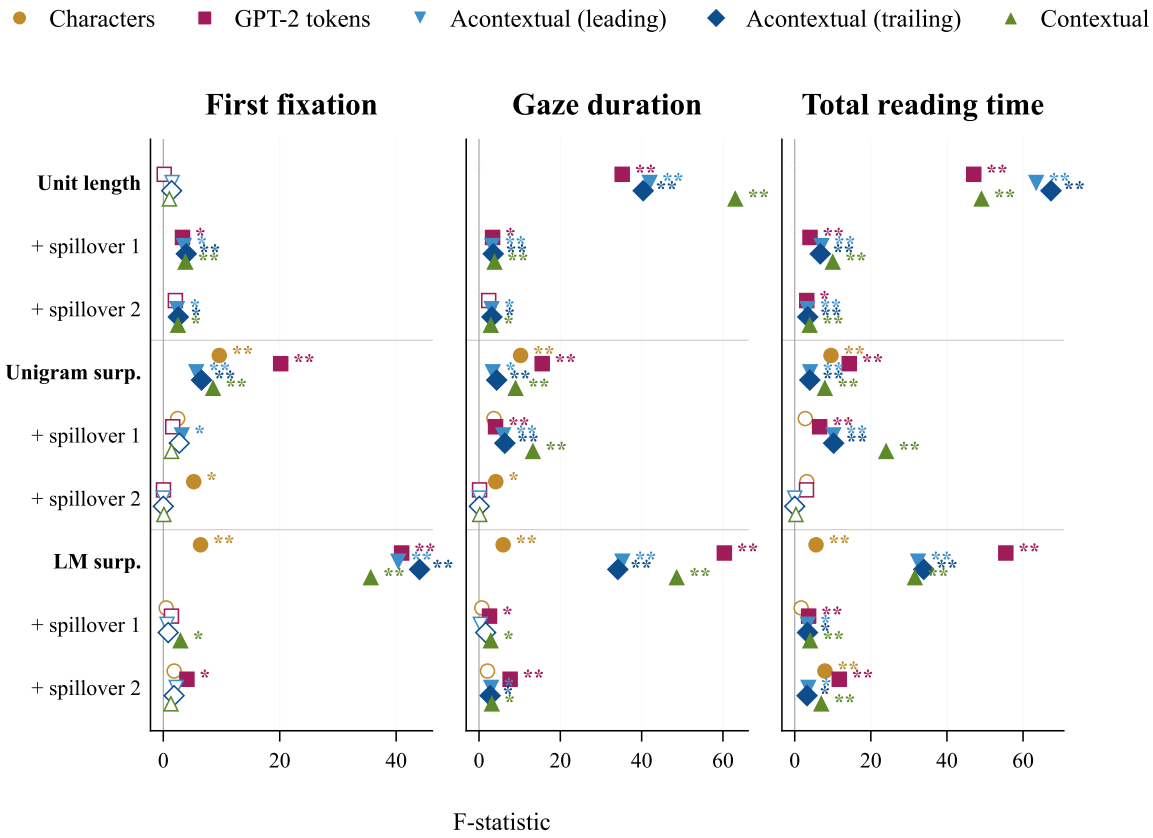

Figure 11: Approximate F-statistics for fixed-effect smooth terms from the full-data GAMM fit, grouped by predictor type. Within each group, rows correspond to the current unit, spillover 1, and spillover 2. Filled markers indicate significance ($^{*}\ p < 0.05$; $^{**}\ p < 0.01$).