Surprisal Estimators for Human Reading Times Need Character Models

Byung-Doh Oh **Christian Clark** **William Schuler**
Department of Linguistics
The Ohio State University
{oh.531, clark.3664, schuler.77}@osu.edu

Abstract

While the use of character models has been popular in NLP applications, it has not been explored much in the context of psycholinguistic modeling. This paper presents a character model that can be applied to a structural parser-based processing model to calculate word generation probabilities. Experimental results show that surprisal estimates from a structural processing model using this character model deliver substantially better fits to self-paced reading, eye-tracking, and fMRI data than those from large-scale language models trained on much more data. This may suggest that the proposed processing model provides a more humanlike account of sentence processing, which assumes a larger role of morphology, phonotactics, and orthographic complexity than was previously thought.

1 Introduction and Related Work

Expectation-based theories of sentence processing (Hale, 2001; Levy, 2008) posit that processing difficulty is determined by predictability in context. In support of this position, predictability quantified through surprisal has been shown to correlate with behavioral measures of word processing difficulty (Goodkind and Bicknell, 2018; Hale, 2001; Levy, 2008; Shain, 2019; Smith and Levy, 2013). However, surprisal itself makes no representational assumptions about sentence processing, leaving open the question of how best to estimate its underlying probability model.

In natural language processing (NLP) applications, the use of character models has been popular for several years (Al-Rfou et al., 2019; Kim et al., 2016; Lee et al., 2017). Character models have been shown not only to alleviate problems with out-of-vocabulary words but also to embody morphological information available at the subword level. For this reason, they have been extensively used to model morphological processes (Elsner et al., 2019; Kann and Schütze, 2016) or incorporate morphological information into models of syntactic acquisition (Jin et al., 2019). Nonetheless, the use of character models has been slow to catch on in psycholinguistic surprisal estimation, which has recently focused on evaluating large-scale language models that make predictions at the word level (e.g. Futrell et al. 2019; Goodkind and Bicknell 2018; Hale et al. 2018; Hao et al. 2020). This raises the question of whether incorporating character-level information into an incremental processing model will result in surprisal estimates that better characterize predictability in context.

To answer this question, this paper presents a character model that can be used to estimate word generation probabilities in a structural parser-based processing model.[1] The proposed model defines a process of generating a word from an underlying lemma and a morphological rule, which allows the processing model to capture the predictability of a given word form in a fine-grained manner. Regression analyses on self-paced reading, eye-tracking, and fMRI data demonstrate that surprisal estimates calculated from this character-based structural processing model contribute to substantially better fits compared to those calculated from large-scale language models, despite the fact that these other models are trained on much more data and show lower perplexities on test data. This finding deviates from the monotonic relationship between test perplexity and predictive power observed in previous studies (Goodkind and Bicknell, 2018; Wilcox et al., 2020). Furthermore, it suggests that the character-based structural processing model may provide a more humanlike account of processing difficulty and may suggest a larger role of morphology, phonotactics, and orthographic complexity than was previously

[1] Code for model and experiments is available at https://github.com/byungdoh/acl21_semproc.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 3746–3757
August 1–6, 2021. ©2021 Association for Computational Linguistics

thought.

2 Background

The experiments presented in this paper use surprisal predictors (Shannon, 1948) calculated by an incremental processing model based on a left-corner parser (Johnson-Laird, 1983; van Schijndel et al., 2013). This incremental processing model provides a probabilistic account of sentence processing by making a single lexical attachment decision and a single grammatical attachment decision for each input word.

Surprisal. Surprisal can be defined as the negative log ratio of prefix probabilities of word sequences $w_{1..t}$ at consecutive time steps $t-1$ and t:

$$\mathsf{S}(w_t) \overset{\text{def}}{=} -\log \frac{\mathsf{P}(w_{1..t})}{\mathsf{P}(w_{1..t-1})} \tag{1}$$

These prefix probabilities can be calculated by marginalizing over the hidden states q_t of the forward probabilities of an incremental processing model:

$$\mathsf{P}(w_{1..t}) = \sum_{q_t} \mathsf{P}(w_{1..t}\ q_t) \tag{2}$$

These forward probabilities are in turn defined recursively using a transition model:

$$\mathsf{P}(w_{1..t}\ q_t) \overset{\text{def}}{=} \sum_{q_{t-1}} \mathsf{P}(w_t\ q_t \mid q_{t-1}) \cdot \mathsf{P}(w_{1..t-1}\ q_{t-1}) \tag{3}$$

Left-corner parsing. The transition model presented in this paper is based on a probabilistic left-corner parser (Johnson-Laird, 1983; van Schijndel et al., 2013). Left-corner parsers have been used to model human sentence processing because they define a fixed number of decisions at every time step and also require only a bounded amount of working memory, in keeping with experimental observations of human memory limits (Miller and Isard, 1963). The transition model maintains a distribution over possible working memory store states q_t at every time step t, each of which consists of a bounded number D of nested derivation fragments a^d_t/b^d_t. Each derivation fragment spans a part of a derivation tree from some apex node a^d_t lacking a base node b^d_t yet to come. Previous work has shown that large annotated corpora such as the Penn Treebank (Marcus et al., 1993) do not require more than $D = 4$ of such fragments (Schuler et al., 2010).

At each time step, a left-corner parsing model generates a new word w_t and a new store state q_t in two phases (see Figure 1). First, it makes a *lexical* decision ℓ_t regarding whether to use the word to complete the most recent derivation fragment (*match*), or to use the word to create a new preterminal node a_{ℓ_t} (*no-match*). Subsequently, the model makes a *grammatical* decision g_t regarding whether to use a predicted grammar rule to combine the node constructed in the lexical phase a_{ℓ_t} with the next most recent derivation fragment (*match*), or to use the grammar rule to convert this node into a new derivation fragment a_{g_t}/b_{g_t} (*no-match*):[2]

$$\begin{aligned} \mathsf{P}(w_t\ q_t \mid q_{t-1}) = \sum_{\ell_t, g_t} & \mathsf{P}(\ell_t \mid q_{t-1}) \cdot \\ & \mathsf{P}(w_t \mid q_{t-1}\ \ell_t) \cdot \\ & \mathsf{P}(g_t \mid q_{t-1}\ \ell_t\ w_t) \cdot \\ & \mathsf{P}(q_t \mid q_{t-1}\ \ell_t\ w_t\ g_t) \end{aligned} \tag{4}$$

Thus, the parser creates a hierarchically organized sequence of derivation fragments and joins these fragments up whenever expectations are satisfied.

In order to update the store state based on the lexical and grammatical decisions, derivation fragments above the most recent nonterminal node are carried forward, and derivation fragments below it are set to null ($\perp$):

$$\mathsf{P}(q_t \mid \ldots) \overset{\text{def}}{=} \prod_{d'=1}^{D} \begin{cases} [\![a^{d'}_t, b^{d'}_t = a^{d'}_{t-1}, b^{d'}_{t-1}]\!] & \text{if } d' < d \\ [\![a^{d'}_t, b^{d'}_t = a_{g_t}, b_{g_t}]\!] & \text{if } d' = d \\ [\![a^{d'}_t, b^{d'}_t = \perp, \perp]\!] & \text{if } d' > d \end{cases} \tag{5}$$

where the indicator function $[\![\varphi]\!] = 1$ if φ is true and 0 otherwise, and $d = \text{argmax}_{d'}\{a^{d'}_{t-1} {\neq} \perp\} + 1 - m_{\ell_t} - m_{g_t}$. Together, these probabilistic decisions generate the n unary branches and $n-1$ binary branches of a parse tree in Chomsky normal form for an n-word sentence.

3 Model

3.1 Processing Model

The processing model extends the above left-corner parser to maintain lemmatized predicate information by augmenting each preterminal, apex, and base node to consist not only of a syntactic category label c_{p_t}, $c_{a^d_t}$, or $c_{b^d_t}$, but also of a binary *predicate context vector* $\mathbf{h}_{p_t}$, $\mathbf{h}_{a^d_t}$, or $\mathbf{h}_{b^d_t} \in \{0, 1\}^{K+V\cdot K}$, where K is the size of the set of predicate contexts and V is the maximum valence of any syntactic

[2] Johnson-Laird (1983) refers to lexical and grammatical decisions as 'shift' and 'predict' respectively.

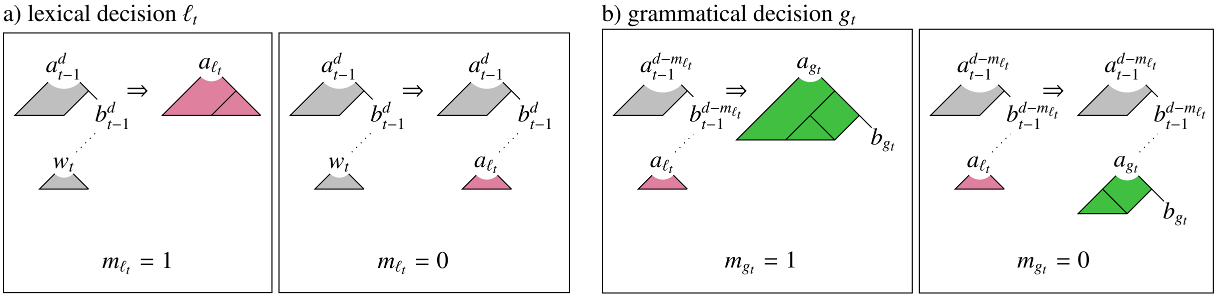


Figure 1: Left-corner parser operations: a) lexical match (m_{ℓ_t}=1) and no-match (m_{ℓ_t}=0) operations, creating new apex a_{ℓ_t}, and b) grammatical match (m_{g_t}=1) and no-match (m_{g_t}=0) operations, creating new apex a_{g_t} and base b_{g_t}.

category.[3] Each 0 or 1 element of this vector represents a unique *predicate context*, which consists of a ⟨*predicate*, *role*⟩ pair that specifies the content constraints of a node in a predicate-argument structure. These predicate contexts are obtained by reannotating the training corpus using a generalized categorial grammar of English (Nguyen et al., 2012),[4] which is sensitive to syntactic valence and non-local dependencies.

Lexical decisions. Each lexical decision of the parser includes a match decision m_{ℓ_t} and decisions about a syntactic category c_{ℓ_t} and a predicate context vector $\mathbf{h}_{\ell_t}$ that together specify a preterminal node p_{ℓ_t}. The probability of generating the match decision and the predicate context vector depends on the base node b^d_{t-1} of the previous derivation fragment (i.e. its syntactic category and predicate context vector). The first term of Equation 4 can therefore be decomposed into the following:

$$\mathsf{P}(\ell_t \mid q_{t-1}) = \underset{m_{\ell_t}\mathbf{h}_{\ell_t}}{\textsc{SoftMax}}(\,\mathrm{FF}_{\theta_\mathrm{L}}[\delta_d{}^\top, [\delta^\top_{c_{b^d_{t-1}}}, \mathbf{h}^\top_{b^d_{t-1}}]\,\mathbf{E}_\mathrm{L}]\,) \cdot \\ \mathsf{P}(c_{\ell_t} \mid q_{t-1}\; m_{\ell_t}\; \mathbf{h}_{\ell_t}) \qquad (6)$$

where FF is a feedforward neural network, and δ_i is a Kronecker delta vector consisting of a one at element i and zeros elsewhere. Depth $d = \operatorname{argmax}_{d'}\{a^{d'}_{t-1} {\neq} \bot\}$ is the number of non-null derivation fragments at the previous time step, and $\mathbf{E}_\mathrm{L}$ is a matrix of jointly trained dense embeddings for each syntactic category and predicate context. The syntactic category and predicate context vector together define a complete preterminal node p_{ℓ_t} for use in the word generation model:

$$p_{\ell_t} \overset{\mathrm{def}}{=} \begin{cases} c_{b^d_{t-1}}, \mathbf{h}_{b^d_{t-1}} + \mathbf{h}_{\ell_t} & \text{if } m_{\ell_t} = 1 \\ c_{\ell_t}, \mathbf{h}_{\ell_t} & \text{if } m_{\ell_t} = 0 \end{cases} \qquad (7)$$

and a new apex node a_{ℓ_t} for use in the grammatical decision model:

$$a_{\ell_t} \overset{\mathrm{def}}{=} \begin{cases} a^d_{t-1} & \text{if } m_{\ell_t} = 1 \\ p_{\ell_t} & \text{if } m_{\ell_t} = 0 \end{cases} \qquad (8)$$

Grammatical decisions. Each grammatical decision includes a match decision m_{g_t} and decisions about a pair of syntactic category labels c_{g_t} and c'_{g_t}, as well as a predicate context composition operator o_{g_t}, which governs how the newly generated predicate context vector $\mathbf{h}_{\ell_t}$ is propagated through its new derivation fragment a_{g_t}/b_{g_t}. The probability of generating the match decision and the composition operators depends on the base node $b^{d-m_{\ell_t}}_{t-1}$ of the previous derivation fragment and the apex node a_{ℓ_t} from the current lexical decision (i.e. their syntactic categories and predicate context vectors). The third term of Equation 4 can accordingly be decomposed into the following:

$$\mathsf{P}(g_t \mid q_{t-1}\; \ell_t\; w_t) = \\ \underset{m_{g_t}o_{g_t}}{\textsc{SoftMax}}(\,\mathrm{FF}_{\theta_\mathrm{G}}[\delta_d{}^\top, [\delta^\top_{c_{b^{d-m_{\ell_t}}_{t-1}}}, \mathbf{h}^\top_{b^{d-m_{\ell_t}}_{t-1}}, \delta^\top_{c_{a_{\ell_t}}}, \mathbf{h}^\top_{a_{\ell_t}}]\,\mathbf{E}_\mathrm{G}]\,) \cdot \\ \mathsf{P}(c_{g_t} \mid q_{t-1}\; \ell_t\; w_t\; m_{g_t}\; o_{g_t}) \cdot \\ \mathsf{P}(c'_{g_t} \mid q_{t-1}\; \ell_t\; w_t\; m_{g_t}\; o_{g_t}\; c_{g_t}) \qquad (9)$$

where $\mathbf{E}_\mathrm{G}$ is a matrix of jointly trained dense embeddings for each syntactic category and predicate context. The composition operators are associated with sparse composition matrices $\mathbf{A}_{o_{g_t}}$ which can be used to compose predicate context vectors associated with the apex node a_{g_t}:

$$a_{g_t} \overset{\mathrm{def}}{=} \begin{cases} a^{d-m_{\ell_t}}_{t-1} & \text{if } m_{g_t} = 1 \\ c_{g_t}, \mathbf{A}_{o_{g_t}}\, \mathbf{h}_{a_{\ell_t}} & \text{if } m_{g_t} = 0 \end{cases} \qquad (10)$$

[3] The valence of a category is the number of unsatisfied syntactic arguments it has. Separate vectors for syntactic arguments are needed in order to correctly model cases such as passives where syntactic arguments do not align with predicate arguments.

[4] The predicates in this annotation scheme come from words that have been lemmatized by a set of rules that have been manually written and corrected in order to account for common irregular inflections.

and sparse composition matrices $\mathbf{B}_{o_{g_t}}$ which can be used to compose predicate context vectors associated with the base node b_{g_t}:

$$b_{g_t} \stackrel{\text{def}}{=} \begin{cases} c'_{g_t}, \mathbf{B}_{o_{g_t}} [\mathbf{h}_{b_{t-1}^{d-m_{\ell_t}}}{}^{\top}, \mathbf{h}_{a_{\ell_t}}{}^{\top}]^{\top} & \text{if } m_{g_t}{=}1 \\ c'_{g_t}, \mathbf{B}_{o_{g_t}} [\mathbf{0}^{\top}, \mathbf{h}_{a_{\ell_t}}{}^{\top}]^{\top} & \text{if } m_{g_t}{=}0 \end{cases} \quad (11)$$

3.2 Character-based Word Model

The baseline version of the word model $\mathsf{P}(w_t \mid q_{t-1}\ \ell_t)$ uses relative frequency estimation with backoff probabilities for out-of-vocabulary words trained using hapax legomena. A character-based test version of this model instead applies a morphological rule r_t to a lemma x_t to generate an inflected form w_t. The set of rules model affixation through string substitution and are inverses of lemmatization rules that are used to derive predicates in the generalized categorial grammar annotation (Nguyen et al., 2012). For example, the rule `%ay`→`%aid` can apply to the word *say* to derive its past tense form *said*. There are around 600 such rules that account for inflection in Sections 02 to 21 of the Wall Street Journal corpus of the Penn Treebank (Marcus et al., 1993), which includes an identity rule for words in bare form and a 'no semantics' rule for generating certain function words.

For an observed input word w_t, the model first generates a list of $\langle x_t, r_t \rangle$ pairs that deterministically generate w_t. This allows the model to capture morphological regularity and estimate how expected a word form is given its predicted syntactic category and predicate context, which have been generated as part of the preceding lexical decision. In addition, this lets the model hypothesize the underlying morphological structure of out-of-vocabulary words and assign probabilities to them. The second term of Equation 4 can thus be decomposed into the following:

$$\begin{aligned} \mathsf{P}(w_t \mid q_{t-1}\ \ell_t) = \sum_{x_t, r_t} & \mathsf{P}(x_t \mid q_{t-1}\ \ell_t) \cdot \\ & \mathsf{P}(r_t \mid q_{t-1}\ \ell_t\ x_t) \cdot \\ & \mathsf{P}(w_t \mid q_{t-1}\ \ell_t\ x_t\ r_t) \end{aligned} \quad (12)$$

The probability of generating the lemma sequence depends on the syntactic category $c_{p_{\ell_t}}$ and predicate context $\mathbf{h}_{\ell_t}$ resulting from the preceding lexical decision ℓ_t:

$$\mathsf{P}(x_t \mid q_{t-1}\ \ell_t) = \prod_i \underset{x_{t,i}}{\textsc{SoftMax}}(\, \mathbf{W}_{\text{X}}\, \mathbf{x}_{t,i} + \mathbf{b}_{\text{X}}\,) \quad (13)$$

where $x_{t,1}, x_{t,2}, ..., x_{t,I}$ is the character sequence of lemma x_t, with $x_{t,1} = \langle s \rangle$ and $x_{t,I} = \langle e \rangle$ as special start and end characters. $\mathbf{W}_{\text{X}}$ and $\mathbf{b}_{\text{X}}$ are respectively a weight matrix and bias vector of a softmax classifier. A recurrent neural network (RNN) calculates a hidden state $\mathbf{x}_{t,i}$ for each character from an input vector at that time step and the hidden state after the previous character $\mathbf{x}_{t,i-1}$:

$$\mathbf{x}_{t,i} = \text{RNN}_{\theta_{\text{X}}}(\, [\delta_{c_{p_{\ell_t}}}^{\top}, \mathbf{h}_{\ell_t}^{\top}, \delta_{x_{t,i}}^{\top}]\, \mathbf{E}_{\text{X}}, \mathbf{x}_{t,i-1}^{\top}\,) \quad (14)$$

where $\mathbf{E}_{\text{X}}$ is a matrix of jointly trained dense embeddings for each syntactic category, predicate context, and character.

Subsequently, the probability of applying a particular morphological rule to the generated lemma depends on the syntactic category $c_{p_{\ell_t}}$ and predicate context $\mathbf{h}_{\ell_t}$ from the preceding lexical decision as well as the character sequence of the lemma:

$$\mathsf{P}(r_t \mid q_{t-1}\ \ell_t\ x_t) = \underset{r_t}{\textsc{SoftMax}}(\, \mathbf{W}_{\text{R}}\, \mathbf{r}_{t,I} + \mathbf{b}_{\text{R}}\,) \quad (15)$$

where $\mathbf{W}_{\text{R}}$ and $\mathbf{b}_{\text{R}}$ are respectively a weight matrix and bias vector of a softmax classifier. $\mathbf{r}_{t,I}$ is the last hidden state of an RNN that takes as input the syntactic category, predicate context, and character sequence of the lemma $x_{t,2}, x_{t,3}, ..., x_{t,I-1}$ without the special start and end characters:

$$\mathbf{r}_{t,i} = \text{RNN}_{\theta_{\text{R}}}(\, [\delta_{c_{p_{\ell_t}}}^{\top}, \mathbf{h}_{\ell_t}^{\top}, \delta_{x_{t,i}}^{\top}]\, \mathbf{E}_{\text{R}}, \mathbf{r}_{t,i-1}^{\top}\,) \quad (16)$$

where $\mathbf{E}_{\text{R}}$ is a matrix of jointly trained dense embeddings for each syntactic category, predicate context, and character.

Finally, as the model calculates probabilities only for $\langle x_t, r_t \rangle$ pairs that deterministically generate w_t, the word probability conditioned on these variables $\mathsf{P}(w_t \mid q_{t-1}\ \ell_t\ x_t\ r_t)$ is deterministic.

4 Experiment 1: Effect of Character Model

In order to assess the influence of the character-based word generation model over the baseline word generation model on the predictive quality of surprisal estimates, linear mixed-effects models containing common baseline predictors and one or more surprisal predictors were fitted to self-paced reading times. Subsequently, a series of likelihood ratio tests were conducted in order to evaluate the relative contribution of each surprisal predictor to regression model fit.

4.1 Response Data

The first experiment described in this paper used the Natural Stories Corpus (Futrell et al., 2018), which contains self-paced reading times from 181 subjects that read 10 naturalistic stories consisting of 10,245 tokens. The data were filtered to exclude observations corresponding to sentence-initial and sentence-final words, observations from subjects who answered fewer than four comprehension questions correctly, and observations with durations shorter than 100 ms or longer than 3000 ms. This resulted in a total of 768,584 observations, which were subsequently partitioned into an exploratory set of 383,906 observations and a held-out set of 384,678 observations. The partitioning allows model selection (e.g. making decisions about predictors and random effects structure) to be conducted on the exploratory set and a single hypothesis test to be conducted on the held-out set, thus eliminating the need for multiple trials correction. All observations were log-transformed prior to model fitting.

4.2 Predictors

The baseline predictors commonly included in all regression models are word length measured in characters and index of word position within each sentence.[5] In addition to the baseline predictors, surprisal predictors were calculated from two variants of the processing model in which word generation probabilities $\mathsf{P}(w_t \mid q_{t-1}\ \ell_t)$ are calculated using relative frequency estimation (*FreqWSurp*) and using the character-based model described in Section 3.2 (*CharWSurp*). Both variants of the processing model were trained on a generalized categorial grammar (Nguyen et al., 2012) reannotation of Sections 02 to 21 of the Wall Street Journal (WSJ) corpus of the Penn Treebank (Marcus et al., 1993). Beam search decoding with a beam size of 5,000 was used to estimate prefix probabilities and surprisal predictors for both variants.

To account for the time the brain takes to process and respond to linguistic input, it is standard practice in psycholinguistic modeling to include 'spillover' variants of predictors from preceding words (Rayner et al., 1983; Vasishth, 2006). However, as including multiple spillover variants of predictors leads to identifiability issues in mixed-effects modeling (Shain and Schuler, 2019), *CharWSurp* and *FreqWSurp* were both spilled over by one position. All predictors were centered and scaled prior to model fitting, and all regression models included by-subject random slopes for all fixed effects as well as random intercepts for each word and subject-sentence interaction, following the convention of keeping the random effects structure maximal in psycholinguistic modeling (Barr et al., 2013).

Model comparison	χ^2	p-value
Full vs. No *CharWSurp*	204.48	0.0001***
Full vs. No *FreqWSurp*	0.024	0.8779

Table 1: Likelihood ratio test evaluating the contribution of *CharWSurp* and *FreqWSurp* in predicting self-paced reading times from the Natural Stories Corpus.

4.3 Likelihood Ratio Testing

A total of three linear mixed-effects models were fitted to reading times in the held-out set using `lme4` (Bates et al., 2015); the full model included the fixed effects of both *CharWSurp* and *FreqWSurp*, and the two ablated models included the fixed effect of either *CharWSurp* or *FreqWSurp*. This resulted in two pairs of nested models whose fit could be compared through a likelihood ratio test (LRT). The first LRT tested the contribution of *CharWSurp* by comparing the fit of the full regression model to that of the regression model without the fixed effect of *CharWSurp*. Similarly, the second LRT tested the contribution of *FreqWSurp* by comparing the fit of the full regression model to that of the regression model without its fixed effect.

4.4 Results

The results in Table 1 show that the contribution of *CharWSurp* in predicting reading times is statistically significant over and above that of *FreqWSurp* ($p < 0.0001$), while the converse is not significant ($p = 0.8779$). This demonstrates that incorporating a character-based word generation model to the structural processing model better captures predictability in context, subsuming the effects of the processing model without it.

5 Experiment 2: Comparison to Other Models

To further examine the impact of the character-based word generation model, *CharWSurp* and *Fre-*

[5] Although unigram surprisal or 5-gram surprisal is also commonly included as a baseline predictor, it was not included in this experiment due to convergence issues.

qWSurp were evaluated against surprisal predictors calculated from a number of other large-scale pretrained language models and smaller parser-based models. To compare the predictive power of surprisal estimates from different language models on equal footing, we calculated the increase in log-likelihood (ΔLL) to a baseline regression model as a result of including a surprisal predictor, following recent work (Goodkind and Bicknell, 2018; Hao et al., 2020).

5.1 Surprisal Estimates from Other Models

A total of three pretrained language models were used to calculate surprisal estimates at each word.[6]

- *GLSTMSurp* (Gulordava et al., 2018): A two-layer LSTM model trained on ~80M tokens of the English Wikipedia.
- *JLSTMSurp* (Jozefowicz et al., 2016): A two-layer LSTM model with CNN character inputs trained on ~800M tokens of the 1B Word Benchmark (Chelba et al., 2014).
- *GPT2Surp* (Radford et al., 2019): GPT-2 XL, a 48-layer decoder-only transformer model trained on the WebText dataset (~8M web documents).

In addition, three incremental parsing models were used to calculate surprisal estimates:

- *RNNGSurp* (Hale et al., 2018; Dyer et al., 2016): An LSTM-based model with explicit phrase structure, trained on Sections 02 to 21 of the WSJ corpus.
- *vSLCSurp* (van Schijndel et al., 2013): A left-corner parser based on a PCFG with subcategorized syntactic categories (Petrov et al., 2006), trained on a generalized categorial grammar reannotation of Sections 02 to 21 of the WSJ corpus.
- *JLCSurp* (Jin and Schuler, 2020): A neural left-corner parser based on stack LSTMs (Dyer et al., 2015), trained on Sections 02 to 21 of the WSJ corpus.

5.2 Procedures

The set of self-paced reading times from the Natural Stories Corpus after applying the same data exclusion criteria as Experiment 1 provided the response variable for the regression models. In addition to the full dataset, regression models were also fitted to a 'no out-of-vocabulary (No-OOV)' version of the dataset, in which observations corresponding to out-of-vocabulary words for the LSTM language model with the smallest vocabulary (i.e. Gulordava et al., 2018) were also excluded. This exclusion criterion was included in order to avoid putting the LSTM language models that may have unreliable surprisal estimates for out-of-vocabulary words at an unfair disadvantage. This resulted in a total of 744,607 observations in the No-OOV dataset, which were subsequently partitioned into an exploratory set of 371,937 observations and a held-out set of 372,670 observations. All models were fitted to the held-out set, and all observations were log-transformed prior to model fitting.

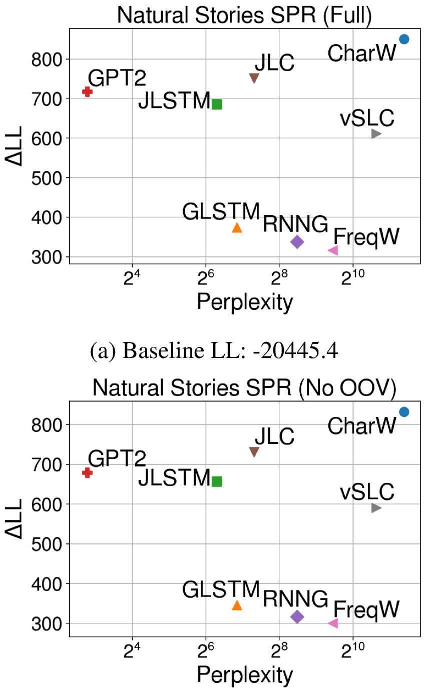
The predictors included in the baseline linear mixed-effects model were word length, word position in sentence, and unigram surprisal. Unigram surprisal was calculated using the KenLM toolkit (Heafield et al., 2013) with parameters trained on the Gigaword 4 corpus (Parker et al., 2009). In order to calculate the increase in log-likelihood (ΔLL) attributable to each surprisal predictor, a 'full' linear-mixed effects model, which includes one surprisal predictor on top of the baseline model, was fitted for each surprisal predictor. As with Experiment 1, the surprisal predictors were spilled over by one position. All predictors were centered and scaled prior to model fitting, and all regression models included by-subject random slopes for all fixed effects and random intercepts for each word and subject-sentence interaction.

Additionally, in order to examine whether any of the models fail to generalize across domains, their perplexity on the entire Natural Stories Corpus was also calculated.

5.3 Results

The results show that surprisal from the character-based structural model (*CharWSurp*) made the biggest contribution to model fit compared to surprisal from other models on both full and No-OOV sets of self-paced reading times (Figure 2; the difference between the model with *CharWSurp* and other models is significant with $p < 0.001$ by a paired permutation test using by-item errors). The exclusion of OOV words did not make a notable difference in the overall trend of ΔLL across models. This finding, despite the fact that the pretrained language models were trained on much

[6] Please refer to the appendix for surprisal calculation, out-of-vocabulary handling, and re-initialization procedures.



(a) Baseline LL: -20445.4

(b) Baseline LL: -17485.2

Figure 2: Perplexity measures from each model, and improvements in regression model log-likelihood from including each surprisal estimate on Natural Stories self-paced reading data.

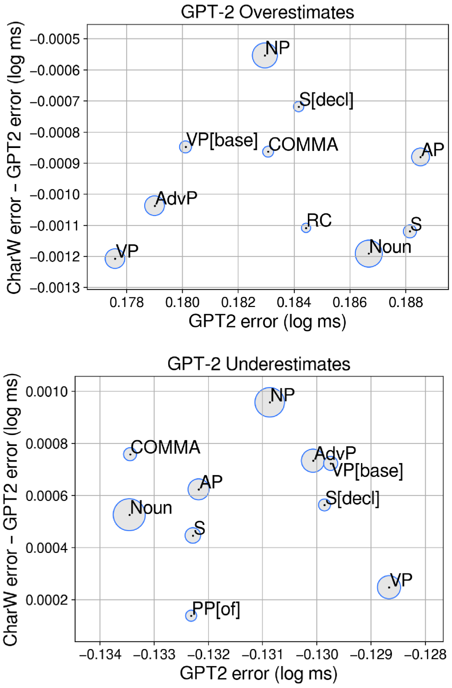


Figure 3: Residual error from the regression model with *GPT2Surp* and change in error from the regression model with *CharWSurp*. Circle widths show the frequency of each syntactic category in the Natural Stories self-paced reading data.

larger datasets and also show lower perplexities on test data,[7] suggests that this model may provide a more humanlike account of processing difficulty. In other words, accurately predicting the next word alone does not fully explain humanlike processing costs that manifest in self-paced reading times. The analysis of residuals grouped by the lowest base category of the previous time step ($c_{b^d_{t-1}}$) from manual annotations (Shain et al., 2018) shows that the improvement of *CharWSurp* over *GPT2Surp* was broad-based across categories (see Figure 3).

6 Experiment 3: Eye-tracking Data

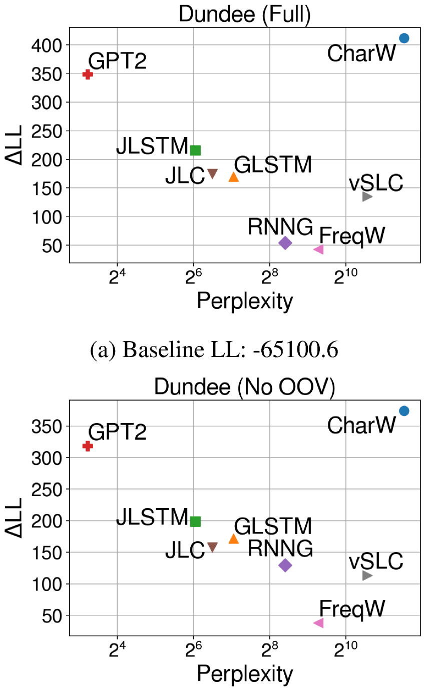
In order to examine whether these results generalize to other latency-based measures, linear-mixed effects models were fitted on the Dundee eye-tracking corpus (Kennedy et al., 2003) to test the contribution of each surprisal predictor, following similar procedures to Experiment 2.

6.1 Procedures

The set of go-past durations from the Dundee Corpus (Kennedy et al., 2003) provided the response variable for the regression models. The Dundee Corpus contains gaze durations from 10 subjects that read 20 newspaper editorials consisting of 51,502 tokens. The data were filtered to exclude unfixated words, words following saccades longer than four words, and words at starts and ends of sentences, screens, documents, and lines. This resulted in the full set with a total of 195,296 observations, which were subsequently partitioned into an exploratory set of 97,391 observations and a held-out set of 97,905 observations. As with Experiment 2, regression models were also fitted to a No OOV version of the dataset, in which observations corresponding to out-of-vocabulary words for the Gulordava et al. (2018) model were also excluded. This resulted in a subset with a total of 184,894 observations (exploratory set of 92,272 observations, held-out set of 92,622 observations). All models were fitted to the held-out set, and all observations were log-transformed prior to model fitting.

The predictors included in the baseline linear mixed-effects models were word length, word position, and saccade length. In order to calculate the increase in log-likelihood from including each surprisal predictor, a full model including one sur-

[7] Perplexity of the parsing models is higher partly because they optimize for a joint distribution over words and trees.



(a) Baseline LL: -65100.6

(b) Baseline LL: -60807.5

Figure 4: Perplexity measures from each model, and improvements in regression model log-likelihood from including each surprisal estimate on Dundee eye-tracking data.

prisal predictor on top of the baseline model was fitted for each surprisal predictor. All surprisal predictors were spilled over by one position, and all predictors were centered and scaled prior to model fitting. All regression models included by-subject random slopes for all fixed effects and random intercepts for each word and sentence.

6.2 Results

The results in Figure 4 show that as with Experiment 2, surprisal from the character-based structural model (*CharWSurp*) made the biggest contribution to model fit on both full and No-OOV sets of go-past durations (the difference between model with *CharWSurp* and other models is significant with $p < 0.001$ by a paired permutation test using by-item errors). In contrast to Natural Stories, surprisal from the two left-corner parsing models (i.e. *vSLCSurp* and *JLCSurp*) did not contribute to as much model fit compared to other models. The exclusion of OOV words again did not make a notable difference in the general trend across different models, although it led to an increase in ΔLL for *GLSTMSurp* and *RNNGSurp*. Residuals grouped by the lowest base category from the previous time

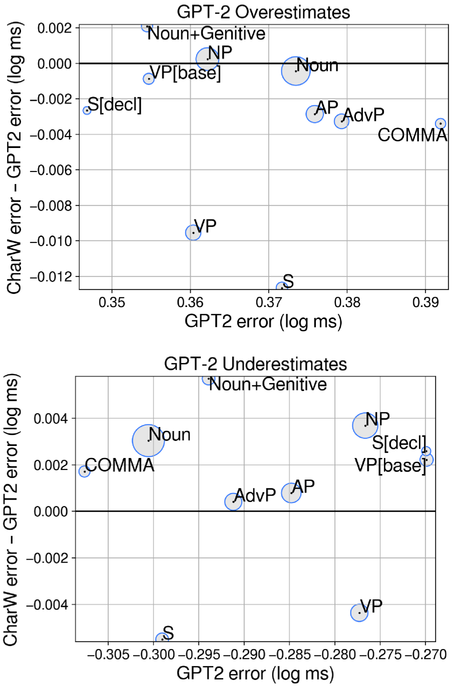


Figure 5: Residual error from the regression model with *GPT2Surp* and change in error from the regression model with *CharWSurp*. Circle widths show the frequency of each syntactic category in the Dundee eye-tracking data.

step show that, similarly to Natural Stories, the improvement of *CharWSurp* over *GPT2Surp* was broad-based across different categories (see Figure 5). These results provide further support for the observation that language models that are trained to predict the next word accurately do not fully explain processing cost in the form of latency-based measures.

7 Experiment 4: fMRI Data

Finally, to examine whether a similar tendency is observed in brain responses, we analyzed the time series of blood oxygenation level-dependent (BOLD) signals in the language network, which were identified using functional magnetic resonance imaging (fMRI). To this end, the novel statistical framework of continuous-time deconvolutional regression (CDR; Shain and Schuler, 2019) was employed. As CDR allows the data-driven estimation of continuous impulse response functions from variably spaced linguistic input, it is more appropriate for modeling fMRI responses, which are typically measured in fixed time intervals. Similarly to the previous experiments, the increase in CDR model log-likelihood as a result of including a surprisal predictor on top of a baseline CDR model was calculated for evaluation.

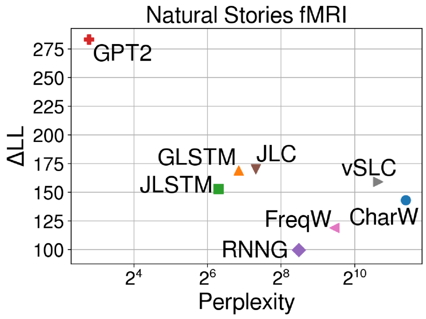
7.1 Procedures

This experiment used the same fMRI data used by Shain et al. (2019), which were collected from 78 subjects that listened to a recorded version of the Natural Stories Corpus. The functional regions of interest (fROI) corresponding to the domain-specific language network were identified for each subject based on the results of a localizer task that they conducted. This resulted in a total of 202,295 observations, which were subsequently partitioned into an exploratory set of 100,325 observations and a held-out set of 101,970 observations by assigning alternate 60-second intervals of BOLD series to different partitions for each participant. All models were fitted to the BOLD signals in the held-out set.

The predictors included in the baseline CDR model were the index of current fMRI sample within the current scan, unigram surprisal, and the deconvolutional intercept which captures the influence of stimulus timing. Following Shain et al. (2019), the CDR models assumed the two-parameter HRF based on the double-gamma canonical HRF (Lindquist et al., 2009). Furthermore, the two parameters of the HRF were tied across predictors, modeling the assumption that the shape of the blood oxygenation response to neural activity is identical in a given region. However, to allow the HRFs to have differing amplitudes, a coefficient that rescales the HRF was estimated for each predictor. The models also included a by-fROI random effect for the amplitude coefficient and a by-subject random intercept.

To calculate the increase in log-likelihood from including each predictor, a full CDR model including the fixed effects of one surprisal predictor was also fitted for each surprisal predictor. All surprisal predictors were included without spillover,[8] and all predictors were centered prior to model fitting.

7.2 Results



(a) Baseline LL: -269825.1

Figure 6: Perplexity measures from each model, and improvements in regression model log-likelihood from including each surprisal estimate on Natural Stories fMRI data.

The results in Figure 6 show that surprisal from GPT-2 (*GPT2Surp*) made the biggest contribution to model fit in comparison to surprisal from other models (difference between model with *GPT2Surp* and other models significant with $p < 0.001$ by a paired permutation test using by-item errors). Most notably, in contrast to self-paced reading times and eye-gaze durations, *CharWSurp* did not contribute as much to model fit on fMRI data, with a ΔLL lower than those of the LSTM language models. This differential contribution of *CharWSurp* across datasets suggests that latency-based measures and blood oxygenation levels may capture different aspects of online processing difficulty.

8 Conclusion

This paper presents a character model that can be used to estimate word generation probabilities in a structural parser-based processing model. Experiments demonstrate that surprisal estimates calculated from this processing model generally contribute to substantially better fits to human response data than those calculated from large-scale pretrained language models or other incremental parsers. These results add a new nuance to the relationship between perplexity and predictive power reported in previous work (Goodkind and Bicknell, 2018; Wilcox et al., 2020). In addition, they suggest that structural parser-based processing models may provide a more humanlike account of sentence processing, and may suggest a larger role of morphology, phonotactics, and orthographic complexity than was previously thought.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported by the National Science Foundation grant #1816891. All views expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.

[8] As CDR estimates continuous HRFs from variably spaced linguistic input, consideration of spillover variants of surprisal predictors was not necessary.

Ethical Considerations

Experiments presented in this work used datasets from previously published research (Futrell et al., 2018; Kennedy et al., 2003; Marcus et al., 1993; Shain et al., 2019), in which the procedures for data collection and validation are outlined.

References

Rami Al-Rfou, Do Kook Choe, Noah Constant, Mandy Guo, and Llion Jones. 2019. Character-level language modeling with deeper self-attention. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pages 3159–3166.

Dale J. Barr, Roger Levy, Christoph Scheepers, and Harry J. Tily. 2013. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language*, 68:255–278.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Phillipp Koehn. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *Proceedings of INTERSPEECH*, pages 2635–2639.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 334–343.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209.

Micha Elsner, Andrea D. Sims, Alexander Erdmann, Antonio Hernandez, Evan Jaffe, Lifeng Jin, Martha Booker Johnson, Shuan Karim, David L. King, Luana Lamberti Nunes, Byung-Doh Oh, Nathan Rasmussen, Cory Shain, Stephanie Antetomaso, Kendra V. Dickinson, Noah Diewald, Michelle McKenzie, and Symon Stevens-Guille. 2019. Modeling morphological learning, typology, and change: What can the neural sequence-to-sequence framework contribute? *Journal of Language Modelling*, 7(1):53–98.

Richard Futrell, Edward Gibson, Harry J. Tily, Idan Blank, Anastasia Vishnevetsky, Steven Piantadosi, and Evelina Fedorenko. 2018. The Natural Stories Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, pages 76–82.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 32–42.

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 8th Workshop on Cognitive Modeling and Computational Linguistics*, pages 10–18.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1195–1205.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, pages 1–8.

John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan Brennan. 2018. Finding syntax in human encephalography with beam search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 2727–2736.

Yiding Hao, Simon Mendelsohn, Rachel Sterneck, Randi Martinez, and Robert Frank. 2020. Probabilistic predictions of people perusing: Evaluating metrics of language model performance for psycholinguistic modeling. In *Proceedings of the 10th Workshop on Cognitive Modeling and Computational Linguistics*, pages 75–86.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696.

Lifeng Jin, Finale Doshi-Velez, Timothy Miller, Lane Schwartz, and William Schuler. 2019. Unsupervised learning of PCFGs with normalizing flow. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2442–2452.

Lifeng Jin and William Schuler. 2020. Memory-bounded neural incremental parsing for psycholinguistic prediction. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 48–61.

Philip N. Johnson-Laird. 1983. *Mental models: Towards a cognitive science of language, inference, and consciousness*. Harvard University Press, Cambridge, MA.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv*.

Katharina Kann and Hinrich Schütze. 2016. MED: The LMU system for the SIGMORPHON 2016 shared task on morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 62–70.

Alan Kennedy, Robin Hill, and Joël Pynte. 2003. The Dundee Corpus. In *Proceedings of the 12th European conference on eye movement*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2741–2749.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Martin A. Lindquist, Ji Meng Loh, Lauren Y. Atlas, and Tor D. Wager. 2009. Modeling the hemodynamic response function in fMRI: Efficiency, bias and mis-modeling. *NeuroImage*, 45(1, Supplement 1):S187–S198.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

George A. Miller and Stephen Isard. 1963. Some perceptual consequences of linguistic rules. *Journal of Verbal Learning and Verbal Behavior*, 2(3):217–228.

Luan Nguyen, Marten van Schijndel, and William Schuler. 2012. Accurate unbounded dependency recovery using generalized categorial grammars. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 2125–2140.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2009. English Gigaword LDC2009T13.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Technical Report*.

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):358–374.

Marten van Schijndel, Andy Exley, and William Schuler. 2013. A model of language processing as hierarchic sequential prediction. *Topics in Cognitive Science*, 5(3):522–540.

William Schuler, Samir AbdelRahman, Tim Miller, and Lane Schwartz. 2010. Broad-coverage incremental parsing using human-like memory constraints. *Computational Linguistics*, 36(1):1–30.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Cory Shain. 2019. A large-scale study of the effects of word frequency and predictability in naturalistic reading. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4086–4094.

Cory Shain, Idan Asher Blank, Marten van Schijndel, William Schuler, and Evelina Fedorenko. 2019. fMRI reveals language-specific predictive coding during naturalistic sentence comprehension. *Neuropsychologia*, 138.

Cory Shain, Marten van Schijndel, and William Schuler. 2018. Deep syntactic annotations for broad-coverage psycholinguistic modeling. In *Workshop on Linguistic and Neuro-Cognitive Resources (LREC 2018)*.

Cory Shain and William Schuler. 2019. Continuous-Time Deconvolutional Regression for Psycholinguistic Modeling. *PsyArXiv*.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423.

Nathaniel J. Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128:302–319.

Mitchell Stern, Daniel Fried, and Dan Klein. 2017. Effective inference for generative neural parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1695–1700.

Shravan Vasishth. 2006. On the proper treatment of spillover in real-time reading studies: Consequences for psycholinguistic theories. In *Proceedings of the International Conference on Linguistic Evidence*, pages 96–100.

Ethan Gotlieb Wilcox, Jon Gauthier, Jennifer Hu, Peng Qian, and Roger P. Levy. 2020. On the predictive power of neural language models for human real-time comprehension behavior. In *Proceedings of the 42nd Annual Meeting of the Cognitive Science Society*, pages 1707–1713.

A Procedures for Surprisal Calculation

- *GLSTMSurp*, *JLSTMSurp*: These models directly estimate $\mathsf{P}(w_t \mid w_{1..t-1})$, which can be used to calculate $\mathsf{S}(w_t) = -\log \mathsf{P}(w_t \mid w_{1..t-1})$.
- *GPT2Surp*: Since GPT-2 relies on byte-pair encoding (Sennrich et al., 2016), negative log probabilities of word pieces corresponding to w_t were added together to calculate $\mathsf{S}(w_t) = -\log \mathsf{P}(w_t \mid w_{1..t-1})$.
- *RNNGSurp*: Since the generative RNNG model defines a joint distribution over words and trees, we marginalize over trees to calculate $\mathsf{P}(w_t \mid w_{1..t-1})$. To keep this tractable, a word-synchronous beam search (Stern et al., 2017) was used with beam size 100, fast-track beam size 5, and word beam size 10.
- *vSLCSurp*, *JLCSurp*: Beam search decoding with a beam size of 5,000 and 2,000 respectively was used to estimate prefix probabilities and surprisal predictors.

B Procedures for Out-of-vocabulary Handling

- *GLSTMSurp*, *JLSTMSurp*, *JLCSurp*: Out-of-vocabulary (OOV) words in the test corpus were replaced with a corresponding "*UNK*" symbol prior to surprisal estimation.
- *GPT2Surp*: Special OOV handling was not necessary because GPT-2 uses byte-pair encoding (Sennrich et al., 2016).
- *RNNGSurp*, *vSLCSurp*: Mapping rules from the Berkeley parser[9] were used to replace OOV words with a set of unknown word classes (e.g. "*UNK-LC-ing*").

C Procedures for Hidden State Re-initialization

- *GLSTMSurp*, *JLSTMSurp*, *GPT2Surp*: The hidden states of these models were re-initialized at the end of every article before making predictions on the next article.
- *RNNGSurp*, *vSLCSurp*, *JLCSurp*: Since these models predict parsing operations while making word predictions, their hidden states were re-initialized after each sentence.

[9] https://github.com/slavpetrov/berkeleyparser