Supplementary material*

1 Alternative reading-time measures

The results presented in the article are based on first-pass reading times, defined as the sum of fixation durations on a word until the first fixation on any other word. The first-pass reading time on word w_t is zero if a later word from the same sentence was fixated before the first fixation on w_t.

To investigate if our conclusions hold for other reading-time measures, we repeated the analysis using two alternatives: *right-bounded* and *go-past* reading time. Right-bounded time is identical to first-pass time except that it includes the durations of fixations on w_t that occur after a regression. That is, it equals the total duration of fixations on w_t until the first fixation on any later word of the sentence. The go-past reading time is defined as the total duration of fixations on *all* words from w_1 (the first word of the sentence) up to w_t, starting from the first fixation on w_t until the first fixation on any later word of the sentence.[1] As was the case for first-pass, the right-bounded and go-past reading times on w_t are zero if a later word from the same sentence was fixated before the first fixation on w_t.

Results for right-bounded and go-past reading times are presented in Figs. 1 and 2, respectively. The first are fully in line with the first-pass results, although the language models account for slightly less variance in right-bounded than in first-pass time. A comparison of the psychologically most accurate ESN

*Supplement to: Frank, S.L. & Bod, R. (in press). Insensitivity of the human sentence-processing system to hierarchical structure. *Psychological Science*.

[1] Note that the go-past reading time on w_t can include durations of fixations on other words, namely w_1 up to w_{t-1}. This may render go-past durations less appropriate for comparison to estimates of w_t's surprisal.

(400-unit) and PSG (level-3 PSG-s) shows that regression model fit increases significantly when the ESN's surprisal estimates are included in a regression that already contains surprisals according to the PSG, but that the reverse is not the case (see Table 1).

The go-past reading times do not fit surprisal values as well as first-pass times: The amount of variance accounted for is much lower (it is no longer significant for the level-1 PSG-a; $\chi^2 = 1.77; p > .18$) and the relation between linguistic and psychological accuracy is much less clear. For Markov models, higher linguistic accuracy even seems to correspond to lower psychological accuracy. Importantly, however, the hierarchical models are not more psychologically accurate than the sequential models. As shown in Table 1, the surprisal estimates by the PSG with highest psychological accuracy (level-3 PSG-s) does not account for variance in reading times over and above the best (i.e., 200-unit) ESN.

Table 1: Test statistics and p-values for nested regression-model tests.

	PSG over ESN		ESN over PSG	
reading-time measure	χ^2	p	χ^2	p
right-bounded	0.71	.40	6.14	.013
go-past	0.08	.78	3.19	.074

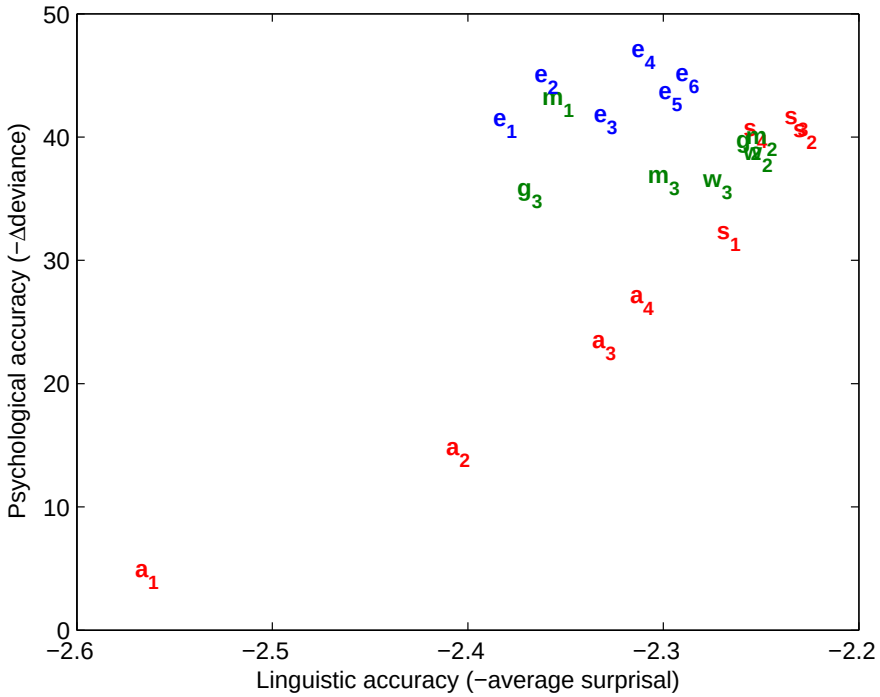

Figure 1: Language model accuracy plotted against psychological accuracy, using right-bounded reading times. a_n is the PSG-a using conditioning information from 1 to n levels up in the parse tree; s_n is the PSG-s using information from 1 to n levels up; m_n, g_n, and w_n are n-th order Markov models with additive, Good-Turing, and Witten-Bell smoothing, respectively; and e_n is the ESN with $100n$ recurrent-layer units.

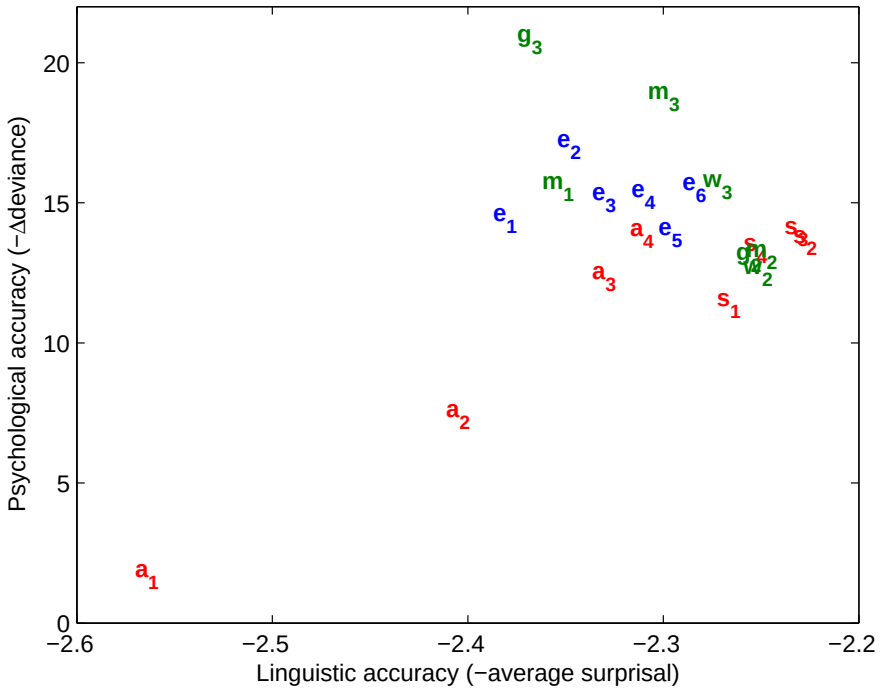

Figure 2: Language model accuracy plotted against psychological accuracy, using go-past reading times.

2 Statistical analysis

To obtain the language models' psychological accuracy measures, linear mixed-effects regression models with subjects and items (i.e., word tokens) as random effects (Baayen, Davidson, & Bates, 2008) were fitted to the reading-time data, using the function lmer in the lme4 package (version 0.999375-32) in R.

2.1 Fitting the baseline models

For each of the three reading-time measures, the baseline model included the seven predictors (main effects) listed in Table 2. In addition, all significant two-way interactions between these predictors were included.[2] All these factors were centered.

Table 3 shows the coefficients of correlation between each pair of fixed-effect factors. A few of these correlations are quite strong, but collinearity between predictors is not very problematic here since the size or significance of coefficients will not be interpreted. Rather, we are only interested in the models' deviance.

Table 2: List of regression predictors.

Abbreviation	Description
pos	position of word in sentence
length	number of letters in word
freq	log of relative word frequency
fwprob	log of forward transitional probability: $\log P(w_t \mid w_{t-1})$
bkprob	log of backward transitional probability: $\log P(w_t \mid w_{t+1})$
prevfix	previous word is fixated
nextfix	next word is fixated

The baseline models also included by-subject and by-item random intervals, and the by-subject random slope that showed the largest effect: This was the factor length for the first-pass and right-bounded measures, and position for the go-past reading times.[3] The fitted model's coefficients are presented in Table 4.

[2] These were determined by first fitting a model with all two-way interactions, and then repeatedly removing the least significant interaction until all were significant ($|t| > 1.96$). Leaving out *all* interactions had little effect on the language models' psychological accuracies.

[3] No additional random slope was included because that would increase computation time

Table 3: Correlation matrix of predictors in the baseline models.

	pos	length	freq	fwprob	bkprob	prevfix	nextfix	pos×length	pos×freq	pos×fwprob	pos×prevfix	pos×nextfix	length×freq	length×fwprob	length×bkprob	length×prevfix	freq×fwprob	freq×bkprob	freq×prevfix	freq×nextfix	fwprob×bkprob	fwprob×prevfix	bkprob×nextfix
length	.03																						
freq	−.04	−.74																					
fwprob	−.03	−.31	.43																				
bkprob	.00	−.32	.44	.15																			
prevfix	−.02	.08	−.07	.32	−.08																		
nextfix	−.00	−.03	.05	−.02	.43	−.03																	
pos×length	.79	.51	−.39	−.17	−.15	.02	−.01																
pos×freq	−.88	−.30	.41	.18	.16	−.01	.02	−.91															
pos×fwprob	−.69	−.20	.27	.60	.08	.20	−.01	−.68	.75														
pos×prevfix	−.56	.04	−.03	.23	−.06	.67	−.02	−.40	.46	.58													
pos×nextfix	−.51	−.03	.06	.00	.30	−.01	.68	−.41	.47	.34	.26												
length×freq	−.04	−.97	.85	.34	.36	−.08	.03	−.51	.35	.23	−.03	.04											
length×fwprob	−.04	−.68	.58	.85	.21	.20	−.01	−.37	.25	.52	.16	.01	.69										
length×bkprob	−.01	−.68	.59	.20	.85	−.09	.33	−.34	.22	.12	−.05	.24	.69	.44									
length×prevfix	−.03	−.36	.26	.46	.08	.81	−.02	−.20	.13	.29	.56	.00	.35	.54	.21								
freq×fwprob	.04	.47	−.63	−.95	−.23	−.25	.00	.26	−.27	−.58	−.19	−.02	−.54	−.90	−.33	−.47							
freq×bkprob	.01	.47	−.63	−.22	−.95	.08	−.37	.24	−.24	−.13	.05	−.27	−.54	−.33	−.91	−.14	.36						
freq×prevfix	.03	.17	−.27	−.48	−.08	−.90	.01	.11	−.13	−.30	−.61	−.01	−.20	−.43	−.11	−.92	.49	.13					
freq×nextfix	.02	.23	−.32	−.09	−.55	.05	−.92	.12	−.13	−.06	.03	−.64	−.26	−.14	−.51	−.06	.16	.57	.06				
fwprob×bkprob	.01	.32	−.44	−.69	−.70	−.17	−.27	.16	−.17	−.40	−.12	−.20	−.36	−.63	−.64	−.33	.71	.72	.34	.40			
fwprob×prevfix	.03	.14	−.22	−.67	−.05	−.80	.02	.10	−.12	−.42	−.55	.00	−.17	−.55	−.08	−.82	.63	.10	.89	.04	.44		
bkprob×nextfix	.01	.20	−.28	−.08	−.71	.05	−.84	.11	−.11	−.05	.03	−.58	−.23	−.12	−.61	−.04	.13	.68	.04	.91	.50	.02	
prevfix×nextfix	.01	−.03	.01	−.17	−.22	−.54	−.60	−.01	−.01	−.11	−.36	−.42	.02	−.11	−.14	−.45	.14	.18	.51	.53	.30	.44	.47

The level of significance (p-value) of each coefficient is estimated by interpreting t-values as z-scores.[4]

2.2 Fitting the surprisal models

Next, for each reading-time measure, 21 more regression models were fitted, each including the surprisal estimates by one of the language models as a fixed effect.[5] Surprisal was not included as a by-subject random slope because of the possibility that participants' sensitivity to surprisal varies more strongly for some sets of surprisal estimates than for others, making the comparisons between language models unreliable. Since subject variability is not currently of interest, it is safer to leave out random surprisal effects.[6]

Finally, two more regression models were fitted, each including surprisal estimates by *both* the PSG and ESN that displayed the highest psychological accuracy. These were the level-3 PSG-s and 400-unit ESN, except when go-past reading times were used, in which case the 200-unit ESN had highest psychological accuracy. The correlations between the corresponding three sets of surprisal estimates, as well as correlations between surprisal and the other predictors, are shown in Table 5. Surprisal turns out to correlate only very weakly with the other predictors but, as expected, different surprisal estimates do correlate quite strongly with each other. The fitted regression models for the different reading-time measures are presented in Table 6.

considerably. Leaving out the single random slope resulted in qualitatively similar psychological accuracies so it is likely the addition of further slopes would not have a large impact.

[4] Considering the very large number of observations (over 190 000), the t- and z-distributions will be nearly identical.

[5] All language models estimated a strictly positive probability for each part-of-speech, so no items needed to be discarded because of infinite surprisal.

[6] This did not negatively affect the significance of the surprisal effect: In the model fitted to first-pass reading times, the surprisal estimates by the level-1 PSG-a also had a significant effect when they were included as a by-subject random slope ($\chi^2(4) = 40.1; p < 10^{-7}$).

Table 4: Fitted baseline models

	First-pass			Right-bounded			Go-past		
Factor	Coef	$\|t\|$	p	Coef	$\|t\|$	p	Coef	$\|t\|$	p
Intercept	221.69	30.66	$< .001$	231.42	29.04	$< .001$	266.36	21.00	$< .001$
pos	−0.27	11.11	$< .001$	−0.28	10.29	$< .001$	0.45	1.05	$> .2$
length	5.65	4.50	$< .001$	7.11	4.77	$< .001$	7.64	17.92	$< .001$
freq	−7.49	36.27	$< .001$	−8.96	38.19	$< .001$	−12.36	25.49	$< .001$
fwprob	1.92	12.86	$< .001$	2.81	16.70	$< .001$	7.18	21.36	$< .001$
bkprob	1.55	10.14	$< .001$	1.95	11.31	$< .001$	2.34	6.54	$< .001$
prevfix	−26.76	50.61	$< .001$	−34.21	58.91	$< .001$	−65.44	52.36	$< .001$
nextfix	−11.41	20.93	$< .001$	−13.04	21.81	$< .001$	−12.37	9.64	$< .001$
pos×length	−0.04	2.66	.008						
pos×freq	−0.04	2.41	.016				−0.14	5.80	$< .001$
pos×fwprob	0.04	3.22	.001	0.04	3.92	$< .001$	0.25	9.08	$< .001$
pos×prevfix							−1.35	12.11	$< .001$
pos×nextfix				−0.16	3.34	$< .001$	−0.64	6.11	$< .001$
length×freq	−2.18	33.70	$< .001$	−2.68	35.62	$< .001$	−3.11	20.40	$< .001$
length×fwprob				0.31	4.12	$< .001$	0.66	5.15	$< .001$
length×bkprob	0.18	2.84	.005	0.25	3.37	$< .001$			
length×prevfix				−4.16	17.07	$< .001$	−5.04	7.23	$< .001$
freq×fwprob	−0.30	5.28	$< .001$	−0.43	5.34	$< .001$			
freq×bkprob	−0.42	5.70	$< .001$	−0.43	5.18	$< .001$	−0.52	3.53	$< .001$
freq×prevfix							3.22	4.39	$< .001$
freq×nextfix	1.55	6.39	$< .001$	1.70	6.36	$< .001$	3.20	5.61	$< .001$
fwprob×bkprob							−0.45	3.73	$< .001$
fwprob×prevfix	4.37	20.48	$< .001$	3.00	11.87	$< .001$	−3.98	6.89	$< .001$
bkprob×nextfix	1.61	6.23	$< .001$	1.56	5.50	$< .001$	−2.30	3.78	$< .001$
prevfix×nextfix	4.51	4.77	$< .001$	6.13	5.92	$< .001$	8.04	3.55	$< .001$

Table 5: Coefficients of correlation between surprisal estimates and other predictors, and between the compared surprisals

	Surprisal estimates by		
	PSG	ESN-400	ESN-200
pos	.00	−.00	−.01
length	−.05	−.05	−.05
freq	.07	.07	.08
fwprob	.06	.07	.07
bkprob	.07	.07	.08
prevfix	.05	.06	.06
nextfix	.02	.02	.02
surprisal PSG		.89	.87
pos×length	−.02	−.02	−.02
pos×freq	.02	.03	.03
pos×fwprob	.03	.03	.04
pos×prevfix	.03	.03	.04
pos×nextfix	.02	.02	.02
length×freq	.07	.07	.07
length×fwprob	.08	.08	.09
length×bkprob	.08	.08	.09
length×prevfix	.06	.06	.07
freq×fwprob	−.10	−.10	−.11
freq×bkprob	−.10	−.10	−.10
freq×prevfix	−.06	−.07	−.08
freq×nextfix	−.05	−.05	−.05
fwprob×bkprob	−.11	−.11	−.12
fwprob×prevfix	−.06	−.07	−.07
bkprob×nextfix	−.06	−.06	−.06
prevfix×nextfix	−.04	−.04	−.04

Table 6: Fitted models with both PSG- and ESN-based surprisal estimates

	First-pass			Right-bounded			Go-past		
Factor	Coef	$\|t\|$	p	Coef	$\|t\|$	p	Coef	$\|t\|$	p
Intercept	221.50	30.63	$< .001$	231.20	29.00	$< .001$	266.16	20.98	$< .001$
surprisal PSG	0.38	0.98		0.37	0.84	$> .4$	0.24	0.28	$> .9$
surprisal ESN	1.13	2.75	$< .001$	1.15	2.48	.013	1.64	1.79	.074
pos	−0.27	11.19	$< .001$	−0.28	10.36	$< .001$	0.45	1.05	$> .2$
length	5.64	4.49	$< .001$	7.10	4.77	$< .001$	7.61	17.86	$< .001$
freq	−7.61	36.78	$< .001$	−9.08	38.61	$< .001$	−12.49	25.70	$< .001$
fwprob	1.95	13.08	$< .001$	2.84	16.87	$< .001$	7.16	21.31	$< .001$
bkprob	1.52	9.95	$< .001$	1.92	11.13	$< .001$	2.32	6.48	$< .001$
prevfix	−26.94	50.91	$< .001$	−34.38	59.15	$< .001$	−65.65	52.48	$< .001$
nextfix	−11.38	20.88	$< .001$	−13.01	21.77	$< .001$	−12.34	9.61	$< .001$
pos×length	−0.04	2.75	.006						
pos×freq	−0.04	2.47	.014				−0.14	5.79	$< .001$
pos×fwprob	0.04	3.36	$< .001$	0.05	4.05	$< .001$	0.25	9.12	$< .001$
pos×prevfix							−1.35	12.09	$< .001$
pos×nextfix				−0.16	3.39	$< .001$	−0.64	6.14	$< .001$
length×freq	−2.17	33.56	$< .001$	−2.67	35.54	$< .001$	−3.10	20.38	$< .001$
length×fwprob				0.32	4.33	$< .001$	0.64	4.96	$< .001$
length×bkprob	0.19	2.99	.003	0.26	3.51	$< .001$			
length×prevfix				−4.11	16.86	$< .001$	−4.96	7.12	$< .001$
freq×fwprob	−0.25	4.30	$< .001$	−0.37	4.47	$< .001$			
freq×bkprob	−0.38	5.16	$< .001$	−0.39	4.72	$< .001$	−0.48	3.23	.001
freq×prevfix							3.23	4.41	$< .001$
freq×nextfix	1.54	6.33	$< .001$	1.68	6.30	$< .001$	3.18	5.56	$< .001$
fwprob×bkprob							−0.43	3.56	$< .001$
fwprob×prevfix	4.35	20.36	$< .001$	2.99	11.84	$< .001$	−3.98	6.89	$< .001$
bkprob×nextfix	1.66	6.41	$< .001$	1.60	5.64	$< .001$	−2.24	3.67	$< .001$
prevfix×nextfix	4.48	4.74	$< .001$	6.11	5.90	$< .001$	7.95	3.51	$< .001$

3 Language model details

3.1 Phrase-structure grammars

The POS-tag sequences of the Dundee corpus sentences were parsed by an incremental parser (Roark, 2001, 2004) that estimates prefix probabilities $P(w_{1...t})$ at each point t of a sentence. These can easily be turned into surprisal values, since

$$-\log P(w_t|w_{1...t-1}) = \log \frac{P(w_{1...t-1})}{P(w_{1...t})}.$$

The probability of a prefix $w_{1...t}$ is the sum of probabilities of all sentences that start with $w_{1...t}$. The probability of a sentence is the sum of probabilities of all its possible tree structures, and the probability of a tree equals the product of probabilities of all production rules involved in its construction. However, the parser does not take all partial parses into account: To improve efficiency, it discards highly unlikely ones. The probability below which a partial parse is removed is controlled by the 'base beam width' parameter. Since earlier research (Frank, 2009) showed that decreasing the value of this parameter increases both linguistic accuracy and the correlation between surprisal estimates and reading times, the base beam width was set to the very small value of 10^{-18} (the default being 10^{-12}).

When inducing the grammar, the probability of a production rule is estimated from its relative frequency in the WSJ treebank, also taking into account the conditional information used by the particular PSG being learned. The number of estimated parameters equals the number of different production rules, which is about 17 000 for the level-1 PSG-a and increases exponentially as the amount of conditional information grows.

The rule probabilities are smoothed by interpolation with the frequencies of rules that use less conditional information. Additional smoothing is applied by setting aside some probability mass for any possible combination of right-hand-side elements that have occurred with a particular nonterminal (see Charniak, 2000, and Roark, 2004, for details).

3.2 Markov models

In order to make the probability of a sentence's first part-of-speech conditional upon it being sentence initial, a special start-of-sentence symbol was added to

the beginning of each sentence. Also, an end-of-sentence symbol was concatenated to the end of each sentence, so that at each point, the models explicitly estimate the probability that the sentence is over.[7] With 45 different POS-tags, plus the start/end symbol, an n-th order Markov model has 46^{n+1} parameters to estimate.

3.2.1 Additive smoothing

POS-tag probabilities were estimated from their occurrence frequencies in the WSJ corpus, with the addition of a small constant value:

$$P_{\text{add}}(w_t|w_{t-n\ldots t-1}) = \frac{\gamma + \text{freq}(w_{t-n\ldots t})}{\gamma|A| + \text{freq}(w_{t-n\ldots t-1})}, \tag{1}$$

where $\text{freq}(w_{t-n\ldots t})$ denotes the number of occurrences of the POS-tag sequence $w_{t-n\ldots t}$ in the WSJ corpus, $|A| = 46$ is the size of the alphabet A (i.e., the set of part-of-speech types plus the end-of-sentence marker), and parameter γ controls smoothing strength. This parameter's value was varied from 0.02 to 1, resulting in the highest linguistic accuracy at $\gamma = 0.2$. This was the value used for the experiments reported in the article.[8]

3.2.2 Good-Turing smoothing

In Good-Turing smoothing, the observed frequencies $\text{freq}(w_{t-n\ldots t})$ are replaced by adjusted $\text{freq}'(w_{t-n\ldots t})$, which are estimates of the frequencies that would occur in a perfect sample. Even strings that have never been observed will have a strictly positive adjusted frequency. The desired probabilities follow directly from the adjusted frequencies:

$$P_{\text{gt}}(w_t|w_{t-n\ldots t-1}) = \frac{\text{freq}'(w_{t-n\ldots t})}{\text{freq}'(w_{t-n\ldots t-1})}.$$

For the reported experiments, the $\text{freq}'(w_{t-n\ldots t})$ were computed using Gale and Sampson's (1995) Simple Good-Turing method.

[7] Roark's parser does the same for the PSG models.

[8] The same value was used irrespective of the model's order n. Only for the second order model (as well as averaged over the three levels of n) did $\gamma = 0.2$ yield the highest linguistic accuracy. In the first and third order models, accuracies were slightly (just over 10^{-4}) higher for $\gamma = 0.5$ and $\gamma = 0.1$, respectively.

3.2.3 Witten-Bell smoothing

The Witten-Bell method (Witten & Bell, 1991) is one of several back-off smoothing methods in which the nth-order model is interpolated with the model of order $n-1$. Following the presentation by Chen and Goodman (1998):

$$P_{\text{wb}}(w_t|w_{t-n\ldots t-1}) = \lambda_{t-n\ldots t-1}P(w_t|w_{t-n\ldots t-1}) + (1-\lambda_{t-n\ldots t-1})P_{\text{wb}}(w_t|w_{t-n+1\ldots t-1}),$$

where $P(w_t|w_{t-n\ldots t-1})$ is the (unsmoothed) probability estimated from the frequencies in the training data and $\lambda_{t-n\ldots t-1}$ controls the smoothing strength for the context $w_{t-n\ldots t-1}$. These parameters are set to:

$$\lambda_{t-n\ldots t-1} = 1 - \frac{N(w_{t-n\ldots t-1})}{N(w_{t-n\ldots t-1}) + \text{freq}(w_{t-n\ldots t})}.$$

Here, $N(w_{t-n\ldots t-1})$ is the number of different (i.e., unique) POS-tags that directly follow $w_{t-n\ldots t-1}$ in the training data. For $n = 0$ there is no smoothing, so $P_{\text{wb}}(w)$ equals the relative frequency of w in the training corpus.

3.3 Echo state networks

The ESN's output activation vector at time step t is computed from the current network state and input symbol i as follows:

$$\begin{aligned} \mathbf{a}_{\text{out}}(t) &= \mathbf{f}_{\text{out}}(\mathbf{W}_{\text{out}}\mathbf{a}_{\text{rec}}(t) + \mathbf{b}) \\ \mathbf{a}_{\text{rec}}(t) &= \mathbf{f}_{\text{rec}}(\mathbf{W}_{\text{rec}}\mathbf{a}_{\text{rec}}(t-1) + \mathbf{W}_{\text{in}}\mathbf{a}^i_{\text{in}}), \end{aligned}$$

where $\mathbf{a}_{\text{rec}}(t)$ and $\mathbf{a}_{\text{out}}(t)$ are, respectively, the activation vectors of the recurrent and output layers directly after processing the input at t; $\mathbf{a}^i_{\text{in}}$ is the input vector representing symbol i; $\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{rec}}$, and $\mathbf{W}_{\text{out}}$ are the matrices of input, recurrent, and output connection weights; $\mathbf{b}$ is the vector of bias weight for the output units; $\mathbf{f}_{\text{rec}}(\mathbf{x})$ is the logistic function applied elementwise to $\mathbf{x}$; and the output activation function $\mathbf{f}_{\text{out}}$ makes sure that output values are strictly positive (which is required to prevent infinite surprisal values) and sum to one:

$$f_{i,\text{out}}(x_1, \ldots, x_n) = \frac{\max\{x_i, \epsilon\}}{\sum_{j=1}^{n} \max\{x_j, \epsilon\}}, \tag{2}$$

where ϵ is a small but strictly positive value. The ESNs used here are architecturally nearly identical to three-layer SRNs (Elman, 1990). They only differ in

the output activation function and the absence of recurrent-layer biases in the ESN.

At the beginning of each sentence, recurrent-unit activations $\mathbf{a}_{\text{rec}}(0)$ are reset to .5 to ensure that sentences are treated as independent from one another, as do the the PSGs and Markov models.

The ESNs have 45 input units: one for each part-of-speech type. There is one additional output unit, which represents the end-of-sentence marker, so there are 46 outputs in total. Like the PSGs and Markov models, the ESNs therefore also estimate, after each word, the probability that the sentence ends at that point.

The number of recurrent units ranged from 100 to 600. Connection weights $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{rec}}$ were chosen randomly from a uniform distribution centered at zero, after which a random selection of 85% of $\mathbf{W}_{\text{rec}}$'s elements were set to zero because using a sparse matrix $\mathbf{W}_{\text{rec}}$ is known to improve network performance (Jaeger, 2001). After a limited exploration of parameter space, the scaling of input weights (between -5 and $+5$) and recurrent weights (spectral radius of $\mathbf{W}_{\text{rec}}$ is 0.7), and the value of ϵ from Eq. 2 ($\epsilon = 0.001$) were chosen to maximize linguistic accuracy.

In most neural network models of sentence processing, the input vectors $\mathbf{a}^i_{\text{in}}$ are orthogonal. However, it has been shown that ESN performance in the next-symbol prediction task can be improved by unsupervised adaptation of the input representations to the training data (Frank & Čerňanský, 2008; Frank & Jacobsson, 2010). Here, the jth element of the input vector representing POS tag i is set to

$$a^i_{\text{in},j} = \log P(i,j) - \log P(i)P(j),$$

where $P(i)$ is the probability of i and $P(i,j)$ is the probability that POS tags i and j occur adjacently (in either order).[9] These probabilities are estimated from frequencies in the training corpus, with additive smoothing.

The networks are trained to predict the next input symbol at each point in the WSJ sentences, using linear regression as a training procedure, as explained in Jaeger (2001). Let $\mathbf{U} = (u_{it})$ denote the matrix of target outputs, that is, $u_{it} = 1$ if at time step t the ith output unit should be active (and 0 otherwise). All recurrent-layer states $\mathbf{a}_{\text{rec}}$ resulting from processing the training inputs are

[9] Note that, as an abuse of notation, j denotes both POS tags and vector elements.

collected in matrix $\mathbf{A}_{\text{rec}}$. The output-weight matrix now becomes:

$$\mathbf{W}_{\text{out}} = \mathbf{U}\mathbf{A}_{\text{rec}}^{-1},$$

where $\mathbf{A}_{\text{rec}}^{-1}$ is the pseudoinverse of $\mathbf{A}_{\text{rec}}$. The bias vector $\mathbf{b}$ is easily obtained by concatenating a row vector of 1s to $\mathbf{A}_{\text{rec}}$. An ESN with n recurrent units has $46 \times (n + 1)$ estimated parameters (i.e., output weights and biases).

ESNs differ from PSGs and Markov models in that they involve a random element since the input and recurrent connection weights are chosen randomly. To make sure that the ESN results are not just an (un)lucky coincidence, three networks of each size were trained, differing only in the weights $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{rec}}$. Of these three, the two networks with lowest and highest psychological accuracy were discarded, that is, the presented ESN results are based on the surprisal estimates with median fit to the reading-time data.

References

Baayen, R. H., Davidson, D. J., & Bates, D. M. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, *59*, 390–412.

Charniak, E. (2000). A maximum-entropy-insipired parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics* (pp. 132–139).

Chen, S. F., & Goodman, J. (1998). *An empirical study of smoothing techniques for language modeling* (Tech. Rep. No. TR-10-98). Computer Science Group, Harvard University, Cambridge, MA.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Frank, S. L. (2009). Surprisal-based comparison between a symbolic and a connectionist model of sentence processing. In N. A. Taatgen & H. van Rijn (Eds.), *Proceedings of the 31st Annual Conference of the Cognitive Science Society* (pp. 1139–1144). Austin, TX: Cognitive Science Society.

Frank, S. L., & Jacobsson, H. (2010). Sentence processing in echo state networks: a qualitative analysis by finite state machine extraction. *Connection Science*, *22*, 135–155.

Frank, S. L., & Čerňanský, M. (2008). Generalization and systematicity in echo state networks. In B. C. Love, K. McRae, & V. M. Sloutsky (Eds.), *Proceedings of the 30th Annual Conference of the Cognitive Science Society* (pp. 733–738). Austin, TX: Cognitive Science Society.

Gale, W. A., & Sampson, G. (1995). Good-Turing frequency estimation without tears. *Journal of Quantitative Linguistics*, *2*, 217–237.

Jaeger, H. (2001). *The "echo state" approach to analysing and training recurrent neural networks* (Tech. Rep. No. GMD 148). German National Research Institute for Computer Science.

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

Roark, B. (2004). Robust garden path parsing. *Natural Language Engineering*, *10*, 1–24.

Witten, I. H., & Bell, T. C. (1991). The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, *37*, 1085–1094.