Uncertainty reduction as a measure of cognitive processing effort

Stefan L. Frank
University of Amsterdam
Amsterdam, The Netherlands
s.l.frank@uva.nl

Abstract

The amount of cognitive effort required to process a word has been argued to depend on the word's effect on the uncertainty about the incoming sentence, as quantified by the entropy over sentence probabilities. The current paper tests this hypothesis more thoroughly than has been done before by using recurrent neural networks for entropy-reduction estimation. A comparison between these estimates and word-reading times shows that entropy reduction is positively related to processing effort, confirming the entropy-reduction hypothesis. This effect is independent from the effect of surprisal.

1 Introduction

In the field of computational psycholinguistics, a currently popular approach is to account for reading times on a sentence's words by estimates of the amount of information conveyed by these words. Processing a word that conveys more information is assumed to involve more cognitive effort, which is reflected in the time required to read the word.

In this context, the most common formalization of a word's information content is its surprisal (Hale, 2001; Levy, 2008). If word string w_1^t (short for $w_1, w_2, \dots w_t$) is the sentence so far and $P(w_{t+1}|w_1^t)$ the occurrence probability of the next word w_{t+1}, then that word's surprisal is defined as $-\log P(w_{t+1}|w_1^t)$. It is well established by now that word-reading times indeed correlate positively with surprisal values as estimated by any sufficiently accurate generative language model (Boston et al., 2008; Demberg and Keller, 2008; Frank, 2009; Roark et al., 2009; Smith and Levy, 2008).

A lesser known alternative operationalization of a word's information content is based on the uncertainty about the rest of the sentence, quantified by Hale (2003, 2006) as the entropy of the probability distribution over possible sentence structures. The reduction in entropy that results from processing a word is taken to be the amount of information conveyed by that word, and was argued by Hale to be predictive of word-reading time. However, this entropy-reduction hypothesis has not yet been comprehensively tested, possibly because of the difficulty of computing the required entropies. Although Hale (2006) shows how sentence entropy can be computed given a PCFG, this computation is not feasible when the grammar is of realistic size.

Here, we empirically investigate the entropy-reduction hypothesis more thoroughly than has been done before, by using recurrent neural networks as language models. Since these networks do not derive any structure, they provide estimates of *sentence* entropy rather than sentence-*structure* entropy. In practice, these two entropies will generally be similar: If the rest of the sentence is highly uncertain, so is its structure. Sentence entropy can therefore be viewed as a simplification of structure entropy; one that is less theory dependent since it does not rely on any particular grammar. The distinction between entropy over sentences and entropy over structures will simply be ignored in the remainder of this paper.

Results show that, indeed, a significant fraction of variance in reading-time data is accounted for by entropy reduction, over and above surprisal.

2 Entropy and sentence processing

2.1 Sentence entropy

Let W be the set of words in the language and W^i the set of all word strings of length i. The set of complete sentences, denoted $\mathcal{S}$, contains all word strings of any length (i.e., $\bigcup_{i=0}^{\infty} W^i$), except that a special end-of-sentence marker $<$/s$>$ is attached to the end of each string.

Proceedings of the 2010 Workshop on Cognitive Modeling and Computational Linguistics, ACL 2010, pages 81–89,
Uppsala, Sweden, 15 July 2010. ©2010 Association for Computational Linguistics

A generative language model defines a probability distribution over $\mathcal{S}$. The entropy of this distribution is

$$H = -\sum_{w_1^j \in \mathcal{S}} P(w_1^j) \log P(w_1^j).$$

As words are processed one by one, the sentence probabilities change. When the first t words (i.e., the string $w_1^t \in W^t$) of a sentence have been processed, the entropy of the probability distribution over sentences is

$$H(t) = -\sum_{w_1^j \in \mathcal{S}} P(w_1^j | w_1^t) \log P(w_1^j | w_1^t). \quad (1)$$

In order to simplify later equations, we define the function $h(y|x) = -P(y|x) \log P(y|x)$, such that Eq. 1 becomes

$$H(t) = \sum_{w_1^j \in \mathcal{S}} h(w_1^j | w_1^t).$$

If the first t words of w_1^j do not equal w_1^t (or w_1^j has fewer than $t+1$ words),[1] then $P(w_1^j|w_1^t) = 0$ so $h(w_1^j|w_1^t) = 0$. This means that, for computing $H(t)$, only the words from $t+1$ onwards need to be taken into account:

$$H(t) = \sum_{w_{t+1}^j \in \mathcal{S}} h(w_{t+1}^j | w_1^t).$$

The reduction in entropy due to processing the next word, w_{t+1}, is

$$\Delta H(t+1) = H(t) - H(t+1). \quad (2)$$

Note that positive ΔH corresponds to a *decrease* in entropy. According to Hale (2006), the nonnegative reduction in entropy (i.e., $\max\{0, \Delta H\}$) reflects the cognitive effort involved in processing w_{t+1} and should therefore be predictive of reading time on that word.

2.2 Suffix entropy

Computing $H(t)$ is computationally feasible only when there are very few sentences in $\mathcal{S}$, or when the language can be described by a small grammar. To estimate entropy in more realistic situations, an obvious solution is to look only at the next few words instead of all complete continuations of w_1^t.

Let $\mathcal{S}^m$ be the subset of $\mathcal{S}$ containing all (and only) sentences of length m or less, counting also the </s> at the end of each sentence. Note that this set includes the 'empty sentence' consisting of only </s>. The set of length-m word strings that do not end in </s> is W^m. Together, these sets form $\mathcal{W}^m = W^m \cup \mathcal{S}^m$, which contains all the relevant strings for defining the entropy over strings up to length m.[2] After processing w_1^t, the entropy over strings up to length $t+n$ is:

$$H_n(t) = \sum_{w_1^j \in \mathcal{W}^{t+n}} h(w_1^j | w_1^t) = \sum_{w_{t+1}^j \in \mathcal{W}^n} h(w_{t+1}^j | w_1^t).$$

It now seems straightforward to define suffix-entropy reduction by analogy with sentence-entropy reduction as expressed in Eq. 2: Simply replace H by H_n to obtain

$$\Delta H_n^{\text{suf}}(t+1) = H_n(t) - H_n(t+1). \quad (3)$$

As indicated by its superscript label, ΔH_n^{suf} quantifies the reduction in uncertainty about the upcoming n-word suffix. However, this is conceptually different from the original ΔH of Eq. 2, which is the reduction in uncertainty about the identity of the current sentence. The difference becomes clear when we view the sentence processor's task as that of selecting the correct element from $\mathcal{S}$. If this set of complete sentences is approximated by $\mathcal{W}^{t+n}$, and the task is to select one element from that set, an alternative definition of suffix-entropy reduction arises:

$$\begin{aligned}
&\Delta H_n^{\text{sent}}(t+1) \\
&= \sum_{w_1^j \in \mathcal{W}^{t+n}} h(w_1^j | w_1^t) \quad - \sum_{w_1^j \in \mathcal{W}^{t+n}} h(w_1^j | w_1^{t+1}) \\
&= \sum_{w_{t+1}^j \in \mathcal{W}^n} h(w_{t+1}^j | w_1^t) \quad - \sum_{w_{t+2}^j \in \mathcal{W}^{n-1}} h(w_{t+2}^j | w_1^{t+1}) \\
&= H_n(t) - H_{n-1}(t+1). \quad (4)
\end{aligned}$$

The label 'sent' indicates that ΔH_n^{sent} quantifies the reduction in uncertainty about which sentence forms the current input. This uncertainty is approximated by marginalizing over all word strings longer than $t+n$.

It is easy to see that

$$\lim_{n\to\infty} \Delta H_n^{\text{suf}} = \lim_{n\to\infty} \Delta H_n^{\text{sent}} = \Delta H,$$

[1] Since w_1^j ends with $</s>$ and w_1^t does not, the two strings must be different. Consequently, if w_1^j is t words long, then $P(w_1^j|w_1^t) = 0$.

[2] The probability of a string $w_1^m \in W^m$ is the summed probability of all sentences with prefix w_1^m.

so both approximations of entropy reduction appropriately converge to ΔH in the limit. Nevertheless, they formalize different quantities and may well correspond to different cognitive factors. If it is true that cognitive effort is predicted by the reduction in uncertainty about the identity of the incoming sentence, we should find that word-reading times are predicted more accurately by ΔH_n^{sent} than by ΔH_n^{suf}.

2.3 Relation to next-word entropy

In the extreme case of $n = 1$, Eq. 4 reduces to

$$\Delta H_1^{\text{sent}}(t+1) = H_1(t) - H_0(t+1) = H_1(t),$$

so the reduction of entropy over the single next word w_{t+1} equals the next-word entropy just before processing that word. Note that $\Delta H_1^{\text{sent}}(t+1)$ is independent of the word at $t + 1$, making it a severely impoverished measure of the uncertainty reduction caused by that word. We would therefore expect reading times to be predicted more accurately by ΔH_n^{sent} with $n > 1$, and possibly even by ΔH_1^{suf}.

Roark et al. (2009) investigated the relation between $H_1(t + 1)$ and reading time on w_{t+1}, and found a significant positive effect: Larger next-word entropy directly *after* processing w_{t+1} corresponded to longer reading time *on* that word. This is of particular interest because $H_1(t + 1)$ necessarily correlates *negatively* with entropy reduction $\Delta H_n^{\text{sent}}(t + 1)$: If entropy is large after w_{t+1}, chances are that it did not reduce much through processing of w_{t+1}. Indeed, in our data set, $H_1(t + 1)$ and $\Delta H_n^{\text{sent}}(t + 1)$ correlate between $r = -.29$ and $r = -.26$ (for $n = 2$ to $n = 4$) which is highly significantly ($p \approx 0$) different from 0. Roark et al.'s finding of a positive relation between $H_1(t + 1)$ and reading time on w_{t+1} therefore seems to disconfirm the entropy-reduction hypothesis.

3 Method

A set of language models was trained on a corpus of POS tags of sentences. The advantage of using POS tags rather than words is that their probabilities can be estimated much more accurately and, consequently, more accurate prediction of word-reading time is possible (Demberg and Keller, 2008; Roark et al., 2009). Subsequent to training, the models were made to generate estimates of surprisal and entropy reductions ΔH_n^{suf} and ΔH_n^{sent} over a test corpus. These estimates were then compared to reading times measured over the words of the same test corpus. This section presents the data sets that were used, language-model details, and the evaluation metric.

3.1 Data

The models were trained on the POS tag sequences of the full WSJ corpus (Marcus et al., 1993). They were evaluated on the POS-tagged Dundee corpus (Kennedy and Pynte, 2005), which has been used in several studies that investigate the relation between word surprisal and reading time (Demberg and Keller, 2008; Frank, 2009; Smith and Levy, 2008). This 2 368-sentence (51 501 words) collection of British newspaper editorials comes with eye-tracking data of 10 participants. POS tags for the Dundee corpus were taken from Frank (2009).

For each word and each participant, reading time was defined as the total fixation time on that word before any fixation on a later word of the same sentence. Following Demberg and Keller (2008), data points (i.e., word/participant pairs) were removed if the word was not fixated, was presented as the first or last on a line, contained more than one capital letter or a non-letter (e.g., the apostrophe in a clitic), or was attached to punctuation. Mainly due to the large number (over 46%) of nonfixations, 62.8% of data points were removed, leaving 191 380 data points (between 16 469 and 21 770 per participant).

3.2 Language model

Entropy is more time consuming to compute than surprisal, even for $n = 1$, because it requires estimates of the occurrence probabilities at $t + 1$ of *all* word types, rather than just of the actual next word. Moreover, the number of suffixes rises exponentially as suffix length n grows, and, consequently, so does computation time.

Roark et al. (2009) used an incremental PCFG parser to obtain H_1 but this method rapidly becomes infeasible as n grows. Low-order Markov models (e.g., a bigram model) are more efficient and can be used for larger n but they do not form particularly accurate language models. Moreover, Markov models lack cognitive plausibility.

Here, Simple Recurrent Networks (SRNs) (Elman, 1990) are used as language models. When trained to predict the upcoming input in a word sequence, these networks can generate estimates of

$P(w_{t+1}|w_1^t)$ efficiently and relatively accurately. They thereby allow to approximate sentence entropy more closely than the incremental parsers used in previous studies. Unlike Markov models, SRNs have been claimed to form cognitively realistic sentence-processing models (Christiansen and MacDonald, 2009). Moreover, it has been shown that SRN-based surprisal estimates can correlate more strongly to reading times than surprisal values estimated by a phrase-structure grammar (Frank, 2009).

3.2.1 Network architecture and processing

The SRNs comprised three layers of units: the input layer, the recurrent (hidden) layer, and the output layer. Each input unit corresponds to one POS tag, making 45 input units since there are 45 different POS tags in the WSJ corpus. The network's output units represent predictions of subsequent inputs. The output layer also has one unit for each POS tag, plus an extra unit that represents </s>, that is, the absence of any further input. Hence, there were 46 output units. The number of recurrent units was fairly arbitrarily set to 100.

As is common in these networks, the input layer was fully connected to the recurrent layer, which in turn was fully connected to the output layer. Also, there were time-delayed connections from the recurrent layer to itself. In addition, each recurrent and output unit received a bias input.

The vectors of recurrent- and output-layer activations after processing w_1^t are denoted $\mathbf{a}_{\text{rec}}(t)$ and $\mathbf{a}_{\text{out}}(t)$, respectively. At the beginning of each sentence, $\mathbf{a}_{\text{rec}}(0) = 0.5$.

The input vector $\mathbf{a}_{\text{in}}^i$, representing POS tag i, consists of zeros except for a single element (corresponding to i) that equals one. When input i is processed, the recurrent layer's state is updated according to:

$$\mathbf{a}_{\text{rec}}(t) = \mathbf{f}_{\text{rec}}(\mathbf{W}_{\text{rec}}\mathbf{a}_{\text{rec}}(t-1) + \mathbf{W}_{\text{in}}\mathbf{a}_{\text{in}}^i + \mathbf{b}_{\text{rec}}),$$

where matrices $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{rec}}$ contain the network's input and recurrent connection weights, respectively; $\mathbf{b}_{\text{rec}}$ is the vector of recurrent-layer biases; and activation function $\mathbf{f}_{\text{rec}}(\mathbf{x})$ is the logistic function $f(x) = (1+\text{e}^{-x})^{-1}$ applied elementwise to $\mathbf{x}$. The new output vector is now given by

$$\mathbf{a}_{\text{out}}(t) = \mathbf{f}_{\text{out}}(\mathbf{W}_{\text{out}}\mathbf{a}_{\text{rec}}(t) + \mathbf{b}_{\text{out}}),$$

where $\mathbf{W}_{\text{out}}$ is the matrix of output connection weights; $\mathbf{b}_{\text{out}}$ the vector of output-layer biases; and $\mathbf{f}_{\text{out}}(\mathbf{x})$ the softmax function

$$f_{i,\text{out}}(x_1, \ldots, x_{46}) = \frac{\text{e}^{x_i}}{\sum_j \text{e}^{x_j}}.$$

This function makes sure that $\mathbf{a}_{\text{out}}$ sums to one and can therefore be viewed as a probability distribution: The i-th element of $\mathbf{a}_{\text{out}}(t)$ is the SRN's estimate of the probability that the i-th POS tag will be the input at $t+1$, or, in case i corresponds to $<$/s$>$, the probability that the sentence ends after t POS tags.

3.2.2 Network training

Ten SRNs, differing only in their random initial connection weights and biases, were trained using the standard backpropagation algorithm. Each string of WSJ POS tags was presented once, with the sentences in random order. After each POS input, connection weights were updated to minimize the cross-entropy between the network outputs and a 46-element vector that encoded the next input (or marked the end of the sentence) by the corresponding element having a value of one and all others being zero.

3.3 Evaluation

3.3.1 Obtaining surprisal and entropy

Since $\mathbf{a}_{\text{out}}(t)$ is basically the probability distribution $P(w_{t+1}|w_1^t)$, surprisal and H_1 can be read off directly. To obtain H_2, H_3, and H_4, we use the fact that

$$P(w_{t+1}^{t+n}|w_1^t) = \prod_{i=1}^{n} P(w_{t+i}|w_1^{t+i-1}). \qquad (5)$$

Surprisal and entropy estimates were averaged over the ten SRNs. So, for each POS tag of the Dundee corpus, there was one estimate of surprisal and four of entropy (for $n = 1$ to $n = 4$).

Since $H_n(t)$ approximates $H(t)$ more closely as n grows, it would be natural to expect a better fit to reading times for larger n. On the other hand, it goes without saying that H_n is only a very rough measure of a reader's actual uncertainty about the upcoming n inputs, no matter how accurate the language model that was used to compute these entropies. Crucially, the correspondence between H_n and the uncertainty experienced by a reader will grow even weaker with larger n. This is apparent from the fact that, as proven in the Appendix, H_n can be expressed in terms of H_1 and H_{n-1}:

$$H_n(t) = H_1(t) + E(H_{n-1}(t+1)),$$

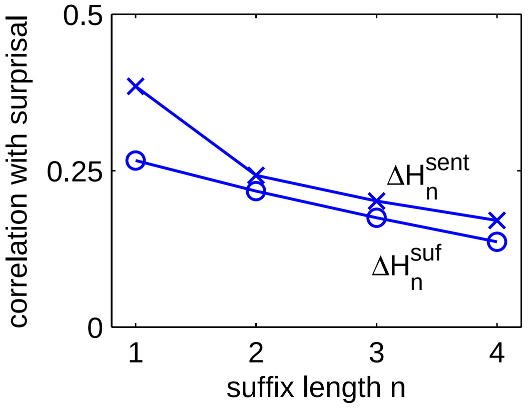

Figure 1: Coefficient of correlation between estimates of surprisal and entropy reduction, as a function of suffix length n.

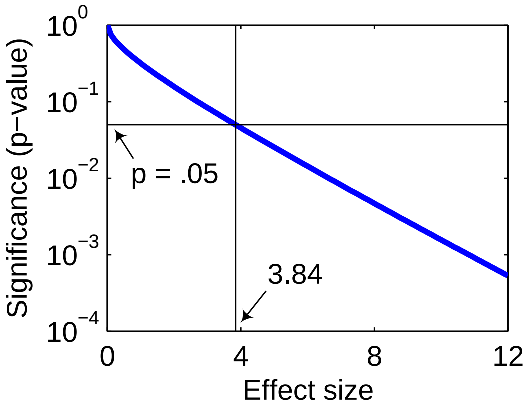

Figure 2: Cumulative χ^2 distribution with 1 degree of freedom, plotting statistical significance (p-value) as a function of effect size.

where $E(x)$ is the expected value of x. Obviously, the expected value of H_{n-1} is less appropriate as an uncertainty measure than is H_{n-1} itself. Hence, H_n can be less accurate than H_{n-1} as a quantification of the actual cognitive uncertainty. For this reason, we may expect larger n to result in *worse* fit to reading-time data.[3]

3.3.2 Negative entropy reduction

Hale (2006) argued for nonnegative entropy reduction $\max\{0, \Delta H\}$, rather than ΔH itself, as a measure of processing effort. For ΔH^{sent}, the difference between the two is negligible because only about 0.03% of entropy reductions are negative. As for ΔH^{suf}, approximately 42% of values are negative so whether these are left out makes quite a difference. Since preliminary experiments showed that word-reading times are predicted much more accurately by ΔH^{suf} than by $\max\{0, \Delta H^{\text{suf}}\}$, only ΔH^{suf} and ΔH^{sent} were used here, that is, negative values were included.

3.3.3 Relation between information measures

Both surprisal and entropy reduction can be taken as measures for the amount of information conveyed by a word, so it is to be expected that they are positively correlated. However, as shown in Figure 1, this correlation is in fact quite weak, ranging from .14 for ΔH_4^{suf} to .38 for ΔH_1^{sent}. In contrast, ΔH_n^{suf} and ΔH_n^{sent} correlate very strongly to each other: The coefficients of correlation range from .73 when $n = 1$ to .97 for $n = 4$.

3.3.4 Fit to reading times

A generalized linear regression model for gamma-distributed data was fitted to the reading times.[4] This model contained several well-known predictors of word-reading time: the number of letters in the word, the word's position in the sentence, whether the next word was fixated, whether the previous word was fixated, log of the word's relative frequency, log of the word's forward and backward transitional probabilities,[5] and surprisal of the part-of-speech. Next, one set of entropy-reduction estimates was added to the regression. The *effect size* is the resulting decrease in the regression model's deviance, which is indicative of the amount of variance in reading time accounted for by those estimates of entropy reduction. Figure 2 shows how effect size is related to statistical significance: A factor forms a significant ($p < .05$) predictor of reading time if its effect size is greater than 3.84.

4 Results and Discussion

4.1 Effect of entropy reduction

Figure 3 shows the effect sizes for both measures of entropy reduction, and their relation to suffix length n. All effects are in the correct direction, that is, larger entropy reduction corresponds to longer reading time. These results clearly support the entropy-reduction hypothesis: A significant

[3] Not to mention the realistic possibility that the cognitive sentence-processing system does not abide by the normative chain rule expressed in Eq. 5.

[4] The reading times, which are approximately gamma distributed, were first normalized to make the scale parameters of the gamma distributions the same across participants.

[5] These are, respectively, the relative frequency of the word given the previous word, and its relative frequency given the next word.

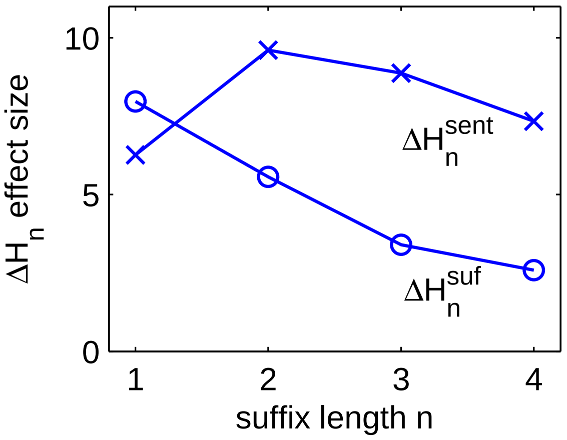

Figure 3: Size of the effect of ΔH_n^{suf} and ΔH_n^{sent} as a function of suffix length n.

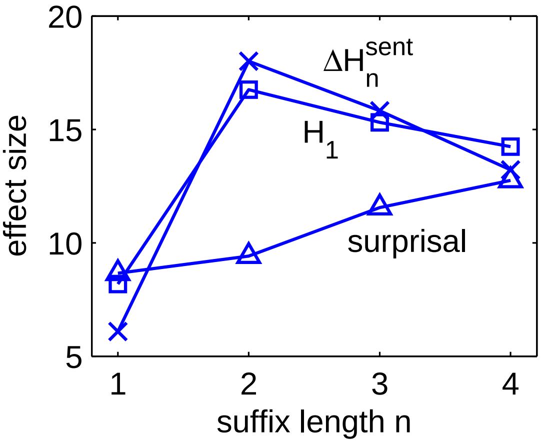

Figure 4: Effect size of entropy reduction (ΔH_n^{sent}), next-word entropy (H_1), or surprisal, over and above the other two predictors.

fraction of variance in reading time is accounted for by the entropy-reduction estimates ΔH_n^{sent}, over and above what is explained by the other factors in the regression analysis, including surprisal. Moreover, the effect of ΔH_n^{sent} is larger than that of ΔH_n^{suf}, indicating that it is indeed uncertainty about the identity of the current sentence, rather than uncertainty about the upcoming input(s), that matters for cognitive processing effort. Only at $n = 1$ was the effect size of ΔH_n^{sent} smaller than that of ΔH_n^{suf}, but it should be kept in mind that ΔH_1^{sent} is independent of the incoming word and is therefore quite impoverished as a measure of the effort involved in processing the word. Moreover, the difference between ΔH_1^{sent} and ΔH_1^{suf} is not significant ($p > .4$), as determined by the bootstrap method (Efron and Tibshirani, 1986). In contrast, the differences are significant when $n > 1$ (all $p < .01$), in spite of the high correlation between ΔH_n^{sent} and ΔH_n^{suf}.

Another indication that cognitive processing effort is modeled more accurately by ΔH_n^{sent} than by ΔH_n^{suf} is that the effect size of ΔH_n^{sent} seems less affected by n. Even though ΔH, the reduction in entropy over complete sentences, is approximated more closely as suffix length grows, increasing n is strongly detrimental to the effect of ΔH_n^{suf}: It is no longer significant for $n > 2$. Presumably, this can be (partly) attributed to the impoverished relation between formal entropy and psychological uncertainty, as explained in Section 3.3.1. In any case, the effect of ΔH_n^{sent} is more stable. Although ΔH_n^{suf} and ΔH_n^{sent} necessarily converge as $n \to \infty$, the two effect sizes seem to diverge up to $n = 3$: The difference between the effect sizes of ΔH_n^{sent} and ΔH_n^{suf} is marginally significantly ($p < .07$) larger for $n = 3$ than for $n = 2$.

4.2 Effects of other factors

It is also of interest that surprisal has a significant effect over and above entropy reduction, in the correct (i.e., positive) direction. When surprisal estimates are added to a regression model that already contains ΔH_n^{sent}, the effect size ranges from 8.7 for $n = 1$ to 13.9 for $n = 4$. This show that there exist independent effects of surprisal and entropy reduction on processing effort.

Be reminded from Section 2.3 that Roark et al. (2009) found a positive relation between reading time on w_{t+1} and $H_1(t+1)$, the next-word entropy after processing w_{t+1}. When that value is added as a predictor in the regression model that already contains surprisal and entropy reduction ΔH_n^{sent}, model fit greatly improves. In fact, as can be seen from comparing Figures 3 and 4, the effect of ΔH_n^{sent} is strengthened by including next-word entropy in the regression model. Moreover, each of the factors surprisal, entropy reduction, and next-word entropy has a significant effect over and above the other two. In all cases, these effects were in the positive direction. This confirms Roark et al.'s finding and shows that it is in fact compatible with the entropy-reduction hypothesis, in contrast to what was suggested in Section 2.3.

5 Discussion and conclusion

The current results contribute to a growing body of evidence that the amount of information conveyed by a word in sentence context is indicative of the amount of cognitive effort required for processing, as can be observed from reading time on the word. Several previous studies have shown that surprisal can serve as a cognitively relevant measure for a word's information content. In contrast, the relevance of entropy reduction as a cognitive measure has not been investigated this thoroughly before. Hale (2003; 2006) presents entropy-reduction accounts of particular psycholinguistic phenomena, but does not show that entropy reduction generally correlates with word-reading times. Roark et al. (2009) presented data that could be taken as evidence against the entropy-reduction hypothesis, but the current paper showed that the next-word entropy effect, found by Roark et al., is independent of the entropy-reduction effect.

It is tempting to take the independent effects of surprisal and entropy reduction as evidence for two distinct cognitive representations or processes, one related to surprisal, the other to entropy reduction. However, it is very well possible that these two information measures are merely complementary formalizations of a single, cognitively relevant notion of word information. Since the quantitative results presented here provide no evidence for either view, a more detailed qualitative analysis is needed.

In addition, the relation between reading time and the two measures of word information may be further clarified by the development of mechanistic sentence-processing models. Both the surprisal and entropy-reduction theories provide only functional-level descriptions (Marr, 1982) of the relation between information content and processing effort, so the question remains which underlying mechanism is responsible for longer reading times on words that convey more information. That is, we are still without a model that proposes, at Marr's computational level, some specific sentence-processing mechanism that takes longer to process a word that has higher surprisal or leads to greater reduction in sentence entropy. For surprisal, Levy (2008) makes a first step in that direction by presenting a mechanistic account of why surprisal would predict word-reading time: If the state of the sentence-processing system is viewed as a probability distribution over all possible interpretations of complete sentences, and processing a word comes down to updating this distribution to incorporate the new information, then the word's surprisal equals the Kullback-Leibler divergence from the old distribution to the new. This divergence is presumed to quantify the amount of work (and, therefore, time) needed to update the distribution. Likewise, Smith and Levy (2008) explain the surprisal effect in terms of a reader's optimal preparation to incoming input. When it comes to entropy reduction, however, no reading-time predicting mechanism has been proposed. Ideally, of course, there should be a single computational-level model that predicts the effects of both surprisal and entropy reduction.

One recent model (Frank, 2010) shows that the reading-time effects of both surprisal and entropy reduction can indeed result from a single processing mechanism. The model simulates sentence comprehension as the incremental and dynamical update of a non-linguistic representation of the state-of-affairs described by the sentence. In this framework, surprisal and entropy reduction are defined with respect to a probabilistic model of the *world*, rather than a model of the *language*: The amount of information conveyed by a word depends on what is asserted by the sentence-so-far, and not on how the sentence's form matches the statistical patterns of the language. As it turns out, word-processing times in the sentence-comprehension model correlate positively with both surprisal and entropy reduction. The model thereby forms a computational-level account of the relation between reading time and both measures of word information. According to this account, the two information measures do not correspond to two distinct cognitive processes. Rather, there is one comprehension mechanism that is responsible for the incremental revision of a mental representation. Surprisal and entropy reduction form two complementary quantifications of the extent of this revision.

Acknowledgments

The research presented here was supported by grant 277-70-006 of the Netherlands Organization for Scientific Research (NWO). I would like to thank Rens Bod, Reut Tsarfaty, and two anonymous reviewers for their helpful comments.

References

M. F. Boston, J. Hale, U. Patil, R. Kliegl, and S. Vasishth. 2008. Parsing costs as predictors of reading difficulty: An evaluation using the Potsdam Sentence Corpus. *Journal of Eye Movement Research*, 2:1–12.

M. H. Christiansen and M. C. MacDonald. 2009. A usage-based approach to recursion in sentence processing. *Language Learning*, 59:129–164.

V. Demberg and F. Keller. 2008. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109:193–210.

B. Efron and R. Tibshirani. 1986. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, 1:54–75.

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

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

S. L. Frank. 2010. The role of world knowledge in sentence comprehension: an information-theoretic analysis and a connectionist simulation. *Manuscript in preparation*.

J. Hale. 2001. A probabilistic Early parser as a psycholinguistic model. In *Proceedings of the second conference of the North American chapter of the Association for Computational Linguistics*, volume 2, pages 159–166. Pittsburgh, PA: Association for Computational Linguistics.

J. Hale. 2003. The information conveyed by words. *Journal of Psycholinguistic Research*, 32:101–123.

J. Hale. 2006. Uncertainty about the rest of the sentence. *Cognitive Science*, 30:643–672.

A. Kennedy and J. Pynte. 2005. Parafoveal-on-foveal effects in normal reading. *Vision Research*, 45:153–168.

R. Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106:1126–1177.

M. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of english: the Penn Treebank. *Computational Linguistics*, 19:313–330.

D. Marr. 1982. *Vision*. San Francisco: W.H. Freeman and Company.

B. Roark, A. Bachrach, C. Cardenas, and C. Pallier. 2009. Deriving lexical and syntactic expectation-based measures for psycholinguistic modeling via incremental top-down parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 324–333. Association for Computational Linguistics.

N. J. Smith and R. Levy. 2008. Optimal processing times in reading: a formal model and empirical investigation. In B. C. Love, K. McRae, and V. M. Sloutsky, editors, *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, pages 595–600. Austin, TX: Cognitive Science Society.

Appendix

It is of some interest that H_n can be expressed in terms of H_1 and the expected value of H_{n-1}. First, note that

$$\begin{aligned} h(w_{t+1}^j|w_1^t) &= -P(w_{t+1}^j|w_1^t)\log P(w_{t+1}^j|w_1^t) \\ &= -P(w_{t+1}|w_1^t)P(w_{t+2}^j|w_1^{t+1})\log\Big(P(w_{t+1}|w_1^t)P(w_{t+2}^j|w_1^{t+1})\Big) \\ &= P(w_{t+2}^j|w_1^{t+1})h(w_{t+1}|w_1^t) + P(w_{t+1}|w_1^t)h(w_{t+2}^j|w_1^{t+1}). \end{aligned}$$

For entropy $H_n(t)$, this makes

$$\begin{aligned} H_n(t) &= \sum_{w_{t+1}^j\in\mathcal{W}^n} h(w_{t+1}^j|w_1^t) \\ &= \sum_{w_{t+1}^j\in\mathcal{W}^n} P(w_{t+2}^j|w_1^{t+1})h(w_{t+1}|w_1^t) && + \sum_{w_{t+1}^j\in\mathcal{W}^n} P(w_{t+1}|w_1^t)h(w_{t+2}^j|w_1^{t+1}) \\ &= \sum_{w_{t+1}\in\mathcal{W}^1}\left(h(w_{t+1}|w_1^t)\sum_{w_{t+2}^j\in\mathcal{W}^{n-1}} P(w_{t+2}^j|w_1^{t+1})\right) && + \sum_{w_{t+1}\in\mathcal{W}^1}\left(P(w_{t+1}|w_1^t)\sum_{w_{t+2}^j\in\mathcal{W}^{n-1}} h(w_{t+2}^j|w_1^{t+1})\right) \\ &= \sum_{w_{t+1}\in\mathcal{W}^1} h(w_{t+1}|w_1^t) && + \sum_{w_{t+1}\in\mathcal{W}^1} P(w_{t+1}|w_1^t)H_{n-1}(t+1) \\ &= H_1(t) + E(H_{n-1}(t+1)). \end{aligned}$$