Fast and Easy: Approximating Uniform Information Density in Language Production

Jesús Calvillo (jesusc@coli.uni-saarland.de)
Saarland University, Germany

Abstract

A model of sentence production is presented, which implements a strategy that produces sentences with more uniform surprisal profiles, as compared to other strategies, and in accordance to the Uniform Information Density Hypothesis (Jaeger, 2006; Levy & Jaeger, 2007). The model operates at the algorithmic level combining information concerning word probabilities and sentence lengths, representing a first attempt to model UID as resulting from underlying factors during language production. The sentences produced by this model showed indeed the expected tendency, having more uniform surprisal profiles and lower average word surprisal, in comparison to other production strategies.

Keywords: information density; sentence production; rational analysis; connectionist; semantics

Introduction

For a given semantics, humans are able to produce a large number of surface representations that express its meaning. However, some constructions are preferred over others, some sentences are easier to understand, while some others are more difficult, so people tend to avoid them.

Uniform Information Density Hypothesis (UID, Jaeger, 2010; Levy & Jaeger, 2007) presents one way to rank sentences according to how uniform their surprisal profiles are; where a sentence is preferred if the surprisal of each of its words remains uniform. This is explained as a rational strategy of language production at the computational level of analysis, as such strategy maximizes the probability of successful communication in a bandwidth-limited noisy channel while maximizing information transmission. Alternatively, and without the assumption of a noisy channel, comprehension effort is also minimized utilizing a UID strategy (Levy & Jaeger, 2007), provided that the effect of surprisal on comprehension effort is superlinear (Hale, 2001; Levy, 2008).

Empirical evidence supports this hypothesis (e.g., Aylett & Turk, 2004; Bell et al., 2003), however, as far as one can tell, no modeling attempts explore this at the algorithmic or implementational levels. Here, a mechanistic account of sentence production is presented, which balances on the one hand speed of information transmission and on the other hand comprehension and production effort. The sentences produced by this strategy present more uniform surprisal profiles, compared to other strategies, and thus, represent a first approximation to UID.

In particular, the model assumes that speakers act under three different pressures: a first one, pushing speakers to be fast under time restrictions; a second one, related to production effort, pushing speakers to produce available content first (see Ferreira & Dell, 2000); and a third one, related to comprehension effort, pushing speakers to avoid high information density structures. Here I present a way to balance these pressures in order to obtain sentences with more uniform surprisal profiles, which could be later linked to a bandwidth-limited communication channel.

The language production model proposed here extends the one presented by Calvillo, Brouwer, and Crocker (2016), which produces sentences describing a given semantics by maximizing word probabilities. The semantic representations used are a variation of those defined by the Distributed Situation Space model (DSS, Frank, Koppen, Noordman, & Vonk, 2003; Frank, Haselager, & van Rooij, 2009). The rest of this section briefly presents the DSS model as well as the model described by Calvillo et al. (2016).

Distributed Situation Space

The DSS model (Frank et al., 2003, 2009) defines a *microworld* in terms of a finite set of *basic events* (e.g., $play(charlie, chess)$) —the smallest meaning-discerning units of propositional meaning in that world. Basic events can be conjoined to form *complex events* (e.g., $play(charlie, chess) \wedge win(charlie)$). However, the microworld poses both hard and probabilistic constraints on event co-occurrence; as a result, some complex events are very common, and some others impossible to happen.

A situation-state space is a large set of m microworld observations defined in terms of n basic events, yielding an $m \times n$ matrix (see Table 1). Each observation in this matrix is encoded by setting basic events that *are the case* in the given observation to 1 (True) and those that are not to 0 (False). This matrix is constructed by sampling m observations such that no observation violates any hard world knowledge constraint, and such that the m observations approximate the probabilistic nature of the microworld. The resulting matrix encodes then *all* knowledge about the microworld, where each column, also called *situation vector*, represents the meaning of each basic event in terms of the observations in which the basic event is true.

Frank et al. (2009) successfully used these DSS representations in a connectionist comprehension model. They defined a microworld consisting of 44 basic events centered around three people. Then they constructed a situation-state space by sampling $25,000$ observations. As an example, in this space the situation vector for $play(charlie, chess)$ would correspond to a column in the matrix, where each dimension corresponds to one observation, and its value would be 1 if Charlie is playing chess in that observation. Finally, they reduced the dimensionality of the resulting $25k$-dimensional situation vectors to 150 dimensions using a competitive layer algorithm.

Table 1: Situation-state space.

	basic event$_1$	basic event$_2$	basic event$_3$	...	basic event$_n$
observation$_1$	1	0	0	...	1
observation$_2$	0	1	1	...	1
observation$_3$	1	1	0	...	0
...	.	.	.	...	.
observation$_m$	0	1	0	...	0

DSS Language Production

DSS representations were also used by Calvillo et al. (2016) in a connectionist model of language production, showing that they are suitable for modeling production.

While Calvillo et al. (2016) used the same microworld as Frank et al. (2009), the DSS representations were modified in order to avoid the competitive layer dimensionality reduction. Instead, the original 25-k dimensional situation vectors were converted to *belief vectors*. Each dimension of the latter is equal to the conditional probability of each basic event given the original 25k-dimensional DSS representation that is associated to each sentence.[1] The result is a 44-dimensional vector that avoids the loss of information associated to the competitive layer algorithm, and consequently renders a higher performance in a language production task.

The architecture of the model presented by Calvillo et al. (2016), represented by the dotted rectangle in Figure 1, implements an extension of a Simple Recurrent Network (Elman, 1990) with a 45-unit input layer, a 120-unit recurrent hidden (htan) layer, and a 43 unit (softmax) output layer. The input layer contains 44 units corresponding to the 44 basic events in the microworld, plus one binary unit indicating whether the model must output an active sentence (1), or a passive one (0). The output layer contains 43 units matching the number of available words in the vocabulary.

Time in the model is discrete. At each time step t, the recurrent hidden layer receives as input the DSS representation, its own activation at time step $t-1$ (zeros at $t=0$) and the identity of the word that was produced at time step $t-1$ (zeros at $t=0$). Activation of the hidden layer is then propagated to the softmax output layer.

The activation of the output layer yields a probability distribution over the available words, where the word produced at time-step t is defined as the one with highest probability (highest activation). Production stops after an end-of-sentence marker has been produced.

[1]This vector is computed by calculating the dot product between the situation-state matrix and the original $25k$-dimensional situation vector, and then normalizing each dimension of the resulting vector by the sum over the dimensions of the original $25k$-dimensional situation vector.

The identity of the word that was produced at time-step $t-1$ is forwarded to the hidden layer through *monitoring* units connecting the output layer to the hidden layer, where only the output unit of the word produced at time-step $t-1$ is activated (set to 1), while all other units are set to 0.

Finally, the hidden and output layers also receive input from a bias unit with a constant activation of 1.

UID Model

The here proposed model architecture, shown in Figure 1, consists of two paths of processing: the first one (above, inside the dotted rectangle), computes word probabilities given the context, and is identical to the model of Calvillo et al. (2016); and the second one (below), receives the output of the former and computes derivation length estimations, i.e., how long a sentence can be if a particular word is produced. We call *probabilities* the layer containing the output of the first path, and *der_lengths* the layer containing the output of the second path.

The output of these two paths is then combined in a final layer (*words*) that receives unmodified copies of the activation of *probabilities* and *der_lengths* and whose activation is a combination of these two types of information. At this point the model produces the word with the highest activation in *words*, whose identity is then passed to the first hidden recurrent layer through monitoring units in order to process the next word production. Finally, production stops when an end-of-sentence marker is produced.

The rest of this section presents in more detail each of these parts, along with their justification.

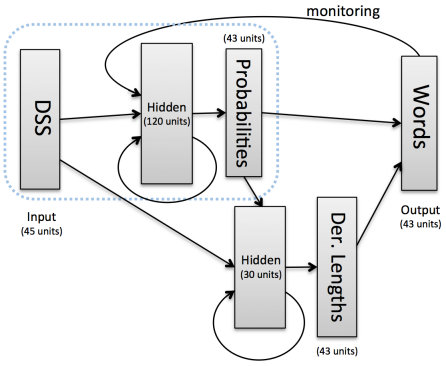

Figure 1: UID Production Model.

Semantic and Linguistic Information

The information content or surprisal of a sentence s is defined as its negative log probability $-logP(s)$. Moreover, sentences express events in the world, such that a sentence can be paired with one or more events, and vice versa. Therefore, we can decompose the probability of a sentence s into:

$$P(s) = \sum_i P(s|e_i)P(e_i)$$

where e_i is an event in the world that is paired with s.

From this, we can distinguish two kinds of information: $P(e_i)$, related to each event that can be paired with the sentence; and $P(s|e_i)$, related to the linguistic elements used in this particular sentence to express e_i.

We call the first one *semantic* surprisal, and the second one *linguistic* surprisal (cf. Frank & Vigliocco, 2011). Semantic surprisal represents how unexpected the events conveyed by the sentence are. Linguistic surprisal can be seen as the information that the sentence conveys, given that the semantics is already known; thus, it is not information about the world, but about the sentence itself.

These two types of information cannot be easily disentangled because they are embedded in each sentence/event. Knowing the identity of an event gives information about the possible related sentences, and vice versa. Nonetheless, based on our definition, we can express total semantic surprisal of a sentence s as:

$$SemSurp(s) = -log\sum_i P(e_i)$$

where e_i is each event that can be expressed by s.

While one sentence can be paired with several events, normally when a speaker produces a sentence, he/she has one specific event in mind e_α. Thus, while total semantic surprisal is as described above, the semantic information/surprisal that the speaker is trying to communicate is only:

$$-logP(e_\alpha)$$

As a result, the relevant information associated with a specific sentence s assuming that the speaker is trying to communicate the event e_α is given by:

$$Surp_{e_\alpha}(s) = -logP(s|e_\alpha)P(e_\alpha)$$

$$= -logP(s|e_\alpha) - logP(e_\alpha)$$

where the semantic information $-logP(e_\alpha)$ remains constant across all different surface realizations that could convey it; in contrast to the linguistic information $-logP(s|e_\alpha)$, which can vary widely depending on the specific syntactic structures or words that the speaker chooses.

Being Easy to Produce

Surprisal Theory (Hale, 2001; Levy, 2008) states that the cognitive effort associated to the processing of a word is proportional to its surprisal. Evidence supporting this has been shown for comprehension (e.g., Hale, 2001; Levy, 2008), and production (e.g., Griffin & Bock, 1998). Therefore, one can assume that a rational model of production would try to minimize effort for both interlocutors.

While comprehension effort is minimized following a UID strategy, production effort can be minimized by following an Availability Based Production strategy (ABP, Ferreira & Dell, 2000), where items are produced as they are available. In this respect, producing the most probable word, and therefore most available, at each time step minimizes (to some extent) production effort by locally minimizing linguistic surprisal:

$$w_{t+1} = \arg\min_w -logP(w|DSS, w_0, ..., w_t)$$

where w is a word in the vocabulary and DSS is the semantic representation related to e_α. This is already implemented by the model described by Calvillo et al. (2016), where the word produced at each time step is the one with highest conditional probability given the semantics and the previously produced words. In our model these probabilities are obtained at the *Probabilities* layer in Figure 1.

Being Fast

The information contained by a sentence results from the sum of the information contained by each of its words. Thus, knowing that the semantic surprisal related to e_α should sum up to $-logP(e_\alpha)$, and that this information is distributed among the words in the sentence, we can calculate average word semantic information/surprisal with respect to e_α:

$$E[WordSemSurp_{e_\alpha}] = \frac{-logP(e_\alpha)}{n}$$

where n is the number of words in the sentence. Hence, if one wants to maximize average semantic information transmission of the desired event e_α, it suffices to minimize n.

We hypothesize that in general speakers tend to maximize information transmission of the desired semantics e_α by minimizing n, and therefore by favoring shorter sentences.

The model presented minimizes sentence lengths by estimating at each time step a score that reflects the expected derivation length that would follow the production of a certain word. This is done by the second path shown in Figure 1, below. This path is constituted by a hidden recurrent layer followed by a softmax layer. The recurrent layer contains 30 sigmoid units and receives as input the DSS semantic representation, the output of *probabilities*, and its own activation at time step $t-1$ (zeros at $t=0$). Activation of this layer is then propagated to a softmax layer (*der_lengths*) with dimensionality equal to the size of the vocabulary(43), and that calculates for each word a probability value *DL*, where values closer to 0 represent longer derivations and values closer to 1 represent shorter derivations, and where probability mass is distributed among all words that can be produced at the given time step. Finally, these layers receive also input from a bias unit with a constant activation of 1.

A model that produces at each time step the word that maximizes this score would prefer words leading to shorter derivations, regardless of their information content:

$$w_{t+1} = \arg\max_w DL(w|DSS, probabilities_{t+1})$$

Being Easy to Comprehend

A model combining the previous two strategies would produce sentences with more uniform surprisal profiles, compared to a model that only applies one of them. However, these strategies do not take into account that world events with high surprisal represent higher comprehension effort.

Speakers know beforehand how unexpected the event they are trying to communicate is. Therefore, one can propose that they balance these two strategies according to this information. That is, when the speaker is trying to communicate an event e_α with low surprisal, the speaker would prefer to be faster; but, when the event represents high surprisal, the speaker would prefer sentences with lower linguistic surprisal and possibly longer. Thus, at each time step, the model would produce the word that maximizes the score:

$$w_{t+1} = \arg\max_w \{(1 - P(e_\alpha))P(w|...) + P(e_\alpha)DL(w|...)\}$$

This final model is expected to produce sentences with more uniform surprisal profiles, compared to strategies that only maximize one of these measures, or that do not take into account semantic surprisal.

In our model this is computed at the *words* layer (see Figure 1), which receives $P(w|...)$ values from the *probabilities* layer and $DL(w|...)$ scores from the *der_lenghts* layer. The value of $P(e_\alpha)$ is assumed to be known.

Training and Evaluation

Examples Set

We use the same examples set as Calvillo et al. (2016), which consist of a set of pairs $\{(DSS_1, \varphi_1), \ldots, (DSS_n, \varphi_n))\}$ where each $DSS_i \in [0,1]^{45}$ is formed by a DSS representation plus an extra bit that indicates whether the model must produce a a passive sentence (0) or an active one (1); and φ_i is the set of all the sentences that encode the information contained in the corresponding DSS_i and in the expected voice.

The sentences are those generated by the microlanguage defined by Frank et al. (2009) (see their Tables 5–8). This microlanguage consists of 40 words that can be combined into 13556 sentences according to its grammar. After adding determiners (a,the) and an end-of-sentence marker (.), there were 43 words, which were encoded at the output layer *probabilities* in the form of localist vectors. After ruling out sentences expressing situations that are not allowed by the microworld, there were a total of 8201 sentences related to 782 DSS representations.

This set was used because it pairs each semantic representation with several sentences, allowing to define different ranking functions. In future work a new set could be defined in order to assess more specific phenomena.

Derivation Length Scores. For each DSS representation, we know beforehand the sentences that can encode it according to the grammar. Furthermore, we know at each derivation point what words can be produced and how long the sentences would be if a particular word is produced. Using this information, we compute a probability distribution over the vocabulary that reflects the length of the sentences that one can expect after producing a particular word.

Given a DSS representation and a derivation point, for each possible word production w_i , we get its minimum derivation length $min\_dl(w_i)$, which is the length of the shortest sentence that can be produced if w_i is produced. Afterwards we calculate a score $dl(w_i)$:

$$dl(w_i) = \max_w \{min\_dl(w)\} - min\_dl(w_i) + 1$$

which is equal to the difference between the greatest min_dl value among all the words that can be currently produced and the min_dl associated to each specific word w_i, plus 1. Finally, in order to have a proper distribution, we normalize by dividing by the sum over all the possible word continuations.

These scores are the values expected at the output layer of *der_lengths*. According to these, all possible word productions at a specific derivation point have some probability mass that is inversely proportional to the length of the shortest sentence that can be obtained by following that production.

Semantic Probability. For each DSS representation in the examples set, a semantic probability value $P(e_\alpha)$ was computed. Considering that the model is trained only on the pairs given in the examples set and that all sentences are presented an equal number of times during training, then the probability of a DSS representation is given by the number of sentences related to that representation divided by the total number of sentences in the examples set.

However, since $P(e_\alpha)$ is used to balance word probabilities and derivation lengths, less biased values are needed because as it is, $P(e_\alpha)$ is in general very low, and $1 - P(e_\alpha)$ is very high. Therefore instead of normalizing by the total number of sentences, normalization is done with respect to the highest number of sentences that can be related to a DSS representation, which is 130. Hence, for each DSS, its probability $P(e_\alpha)$, or henceforth $P(DSS)$, is given by the number of sentences paired with the representation, divided by 130.

Training Procedure

Since the output layer receives unmodified copies from *probabilities* and *der_lengths*, the connections from the latter to the former are fixed one-to-one and do not need training. In other words, the i^{th} unit of *probabilities* is only connected to the i^{th} unit of *words* with a connection weight fixed to 1, and likewise for the connections between *der_lenghts* and *words*.

Prior to training, all weights on the projections between layers (with the exception of those mentioned in the last paragraph) were initialized with random values drawn from a normal distribution $\mathcal{N}(0, 0.1)$. Weights on the bias projections were initially set to zero.

Training consists of setting the connection weights leading to the computation on the one hand of *probabilities* and on the other hand of *der_lengths*, corresponding to the two paths of processing. Accordingly, training is performed in

two phases, in both cases using cross-entropy backpropagation (Rumelhart, Hinton, & Williams, 1986) with weight updates after each word in the sentence of each training item.

probabilities. The first phase corresponds to the training of the path leading to *probabilities*, which is performed as described by Calvillo et al. (2016), where the model is trained to predict the next word given the semantic representation and the previously produced words.

During this phase, the monitoring units were set at time t to what the model was supposed to produce at time $t-1$ (zeros for $t=0$). This reflects the notion that during training the word contained in the training sentence at time-step $t-1$ should be the one informing the next time step, regardless of the previously produced (and possibly different) word. During production, the monitoring units are set to 1.0 for the word that was actually produced and 0.0 everywhere else.

This path was trained for a maximum of 200 epochs, each one consisting of a full presentation of the training set, which was randomized before each epoch. Note that each item of this set consisted of a DSS_i paired with one of the possible sentence realizations describing the state of affairs represented in DSS_i. Hence, during each epoch, the model saw all the possible realizations of DSS_i. An initial learning rate of 0.124 was used, which was halved each time there was no improvement of performance during 15 epochs. No momentum was used. Training halted if the maximum number of epochs was reached or if there was no performance improvement over a 40-epoch interval.

der_lengths. The second path can be trained after the training of the first one is completed. During this phase, the connection weights calculated during the first phase are fixed, so that only the second path weights are modified.

At each time step, the DSS is fed into the first path, which outputs a probability distribution over the vocabulary. This is fed into the second recurrence, as well as the DSS representation. Monitoring units are handled exactly as in the first training phase. The activation of the second recurrence is then propagated to *der_lengths*. Its output is compared to the derivation length values, as defined in the previous section, and finally the connection weights are updated.

Training of this path was performed for a maximum of 80 epochs, with the training items arranged in the same way as in the previous phase. An initial learning rate of 0.24 was used, which was halved each time there was no improvement of performance during 10 epochs. No momentum was used. Training halted if the maximum number of epochs was reached or if there was no performance improvement over a 20-epoch interval.

Evaluation

The model presented defines a production strategy as an interaction between production goals. Thus, in order to assess the model, its productions were compared to those obtained by using the following alternative strategies, where at each time step the model produces the word with:

- Min Linguistic Surprisal
- Min Derivation Length
- Max Word Probability +/* Derivation Length Score
- Complete Model

For each DSS representation in the examples set that was related to more than one sentence (968), the model generated a sentence according to each production strategy.

In order to measure surprisal, a language model was trained implementing a Simple Recurrent Network (Elman, 1990). This model was trained on the whole set of sentences for 200 epochs with a learning rate of 0.24 which was halved each time there was no improvement in performance. Using this language model, surprisal values were calculated for each one of the words of the produced sentences.

Uniformity of information density was measured in terms of standard deviation of word surprisal, assuming that complete uniformity would produce a standard deviation of 0.

Results and Discussion

The results can be seen in Table 2, where the columns denote respectively: production strategy, production accuracy (Acc) as defined by Calvillo et al. (2016) and denoting how precise the sentences convey the given semantics, average sentence length (AvDL), average word surprisal (AvS), and standard deviation of surprisal (Std).

Table 2: Results of each production strategy.

	Acc	AvDL	AvS	Std
Min LS	99.67	9.01	1.0	0.89
Min DL	99.86	**7.55**	1.20	0.97
Max P(+/*)DL	99.82	7.77	1.16	0.95
Max 3P-2DL	98.23	10.15	**0.89**	**0.84**
SemSurp	97.67	10.17	**0.89**	**0.83**

As expected, minimizing linguistic surprisal (Min LS) led to lower surprisal values compared to minimizing derivation lengths (Min DL). Combining these two strategies by a sum or product led to results almost identical to each other, and very close to Min DL, suggesting that derivation length scores were mostly dominating production.

Given that linguistic surprisal and derivation lengths are different in nature, one can expect a more complex relation between them in order for the resulting score to be helpful. Consequently, grid search was performed in order to find linear factors that would minimize the standard deviation of surprisal. The resulting model corresponds to the fourth row in Table 2, where the model produces at each time step the word that maximizes:

$$3P(w|DSS, w0, .., w_n) - 2DL(w|DSS, probabilities)$$

where one can see that minimizing linguistic surprisal is favored, while minimizing derivation lengths is penalized. As a result the sentences produced are longer than only minimizing linguistic surprisal. However, uniformity of information density is higher than with the previous models and additionally average surprisal is lowest.

The final row in Table 2 presents the results of the model that incorporates semantic probabilities. For this case grid search was also used, which led to a model that at each time step produces the word that maximizes:

$$(3.5 - P(DSS))P(w|...) + (P(DSS) - 2.5)DL(w|...)$$

which is very similar to the previous model, but with some influence from semantic probabilities. While the performance of this model is very similar to the previous one, its sentences present slightly higher uniformity of information density; and the influence of semantic surprisal is in the expected direction, where semantics with high surprisal produce longer sentences and vice versa.

The small difference between the last two strategies could be caused by the nature of the language model, which receives no semantic information during training, which means that rather than being a joint model of semantics and sentences, it only considers word sequences. Furthermore, the production model here proposed uses semantic surprisal at a sentence level, while speakers can be sensitive to this information incrementally at a word level. These issues will be addressed in future work.

In general the model outlined here shows: first, that as expected, shorter sentences are more dense in terms of information content. Second, that longer sentences present information in a more uniform way. Third, that sentences with more uniform information densities present in average lower word surprisal, therefore minimizing comprehension effort. And finally and most importantly, that sentences with higher uniformity of information density can be produced by balancing sentence lengths and word probabilities. In future work, this can help to address uniformity for a given channel capacity.

Conclusion

This article presents a model of language production that takes into account word probabilities and sentence lengths in order to produce sentences with uniform surprisal profiles, and in order to model the Uniform Information Density Hypothesis. The sentences produced by this model were compared to those produced using other strategies, showing that the proposed model produces sentences with more uniform surprisal profiles and lower average word surprisal. This model represents a first attempt to model the Uniform Information Density Hypothesis at the algorithmic level, where uniformity arises by balancing word probabilities and sentence lengths in a mechanistic way.

Acknowledgements

I would like to thank Matthew W. Crocker and Harm Brouwer for their helpful comments while writing this article.

References

Aylett, M., & Turk, A. (2004). The smooth signal redundancy hypothesis: A functional explanation for relationships between redundancy, prosodic prominence, and duration in spontaneous speech. *Language and speech*, *47*(1), 31–56.

Bell, A., Jurafsky, D., Fosler-Lussier, E., Girand, C., Gregory, M., & Gildea, D. (2003). Effects of disfluencies, predictability, and utterance position on word form variation in english conversation. *The Journal of the Acoustical Society of America*, *113*(2), 1001–1024.

Calvillo, J., Brouwer, H., & Crocker, M. W. (2016). Connectionist semantic systematicity in language production. In *Proceedings of the 38th annual conference of the cognitive science society.*

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211.

Ferreira, V. S., & Dell, G. S. (2000). Effect of ambiguity and lexical availability on syntactic and lexical production. *Cognitive psychology*, *40*(4), 296–340.

Frank, S., Haselager, W. F. G., & van Rooij, I. (2009). Connectionist semantic systematicity. *Cognition*, *110*(3), 358–379.

Frank, S., Koppen, M., Noordman, L. G. M., & Vonk, W. (2003). Modeling knowledge-based inferences in story comprehension. *Cognitive Science*, *27*(6), 875–910.

Frank, S., & Vigliocco, G. (2011). Sentence comprehension as mental simulation: an information-theoretic perspective. *Information*, *2*(4), 672–696.

Griffin, Z. M., & Bock, K. (1998). Constraint, word frequency, and the relationship between lexical processing levels in spoken word production. *Journal of Memory and Language*, *38*(3), 313 - 338.

Hale, J. (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* (pp. 1–8). Stroudsburg, PA, USA: Association for Computational Linguistics.

Jaeger, T. F. (2006). *Redundancy and syntactic reduction in spontaneous speech*. Unpublished doctoral dissertation, Stanford University.

Jaeger, T. F. (2010). Redundancy and reduction: Speakers manage syntactic information density. *Cognitive psychology*, *61*(1), 23–62.

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

Levy, R., & Jaeger, T. F. (2007). Speakers optimize information density through syntactic reduction. *Advances in neural information processing systems*, *19*, 849.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.