A Liquid-State Model of Variability Effects in Learning Nonadjacent Dependencies

Hartmut Fitz (hartmut.fitz@gmail.com)
Center for Language and Cognition Groningen, University of Groningen
Oude Kijk in 't Jatstraat 26, 9712 EK Groningen, the Netherlands

Abstract

Language acquisition involves learning nonadjacent dependencies that can exist between words in a sentence. Several artificial grammar learning studies have shown that the human ability to detect dependencies between A and B in sequences AXB is influenced by the amount of variation in the X element. This paper presents a model of statistical learning that displays similar behavior on this task and generalizes in a human-like way. The model was also used to predict human behavior for increased distance and more variation in dependencies. We compare this model-based approach with the standard invariance account of the variability effect.

Keywords: Language acquisition; statistical learning; variability; nonlocal dependencies; liquid-state machines.

Introduction

Sentences in natural language are not just sequences of independent words. Dependencies can hold between immediately adjacent words or between words at a distance. Language acquisition involves learning which dependencies are syntactically required to form grammatical sentences. We call a sequence of words in which the final element depends on the identity of the initial element a *frame*. Frames are quite common in natural language. In tense morphology, inflectional morphemes depend on the subject auxiliary, e.g., in 'X **is** VERB-**ing** Y'. The category VERB is highly variable whereas the frame itself is rigid and highly frequent. Nonlocal dependencies are also created by noun-verb number agreement, e.g., in 'the X**s** on the table **are** Y' where dependent elements (plural marker and auxiliary) can be separated by prepositional phrases or relative clauses. Furthermore, it has been argued that frequent three-word frames such as '**You** X **it**' may enable children to induce word categories X and thus solve the bootstrapping problem (Mintz, 2003), in particular if frame elements are function words (Leibbrandt & Powers, 2010). In all these examples, patterns of highly invariant nonadjacent words are separated by highly variable lexical material (fillers).

In the artificial grammar learning paradigm (AGL) several recent studies have investigated how the learning of nonadjacent dependencies is modulated by the amount of variation in the middle slot (Gómez, 2002; Gómez & Maye, 2005; Onnis et al., 2003, 2004). These studies found a *variability effect* in adults and children, and across modalities. In Onnis et al. (2003), for instance, adult subjects were exposed to 432 nonce word strings of the form $A_iX_jB_i$ where $i \in \{1,2,3\}$ and X_j was drawn from sets of various sizes (1, 2, 6, 12, or 24). Subsequently, subjects had to judge the grammaticality of strings that were in the training set (e.g., $A_1X_4B_1$) and of strings in which dependencies were violated in that the final element did not match the initial element (e.g., $A_2X_9B_3$). The results of this experiment are depicted in Figure 1. In conditions of high variability (12 and 24), dependency learning was significantly better than for medium variability (6). The highest accuracy was observed when there was no variation in the middle element (1). Manipulating the amount of variability in the fillers resulted in a U-shaped behavioral pattern.

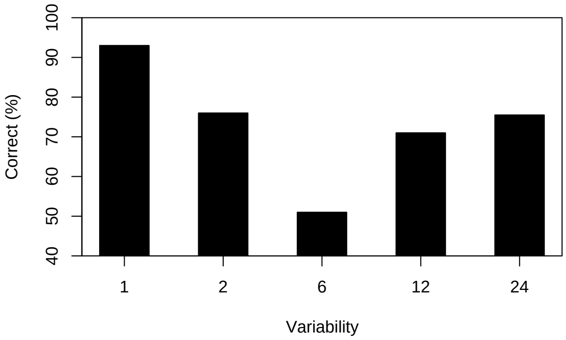

Figure 1: The variability effect in learning nonadjacent dependencies (data reproduced from Onnis et al. 2003).

The experiment was designed to exclude surface distributional properties as explanatory factors. In all variability conditions, for example, each $A..B$ frame occurred the same number of times in training, ruling out a frequency-based account of the variability effect. Other statistical cues such as type frequency and forward transitional probabilities were similarly uninformative (see Figure 2). This suggests that mechanisms of statistical learning which rely on frequency and N-gram information may not be able to account for the learning of nonadjacent dependencies in humans. The only information that was useful in this task was the identity of the frame initial elements. Learners had to attend to these elements and ignore the 'noise' in the middle position. Why did this strategy work

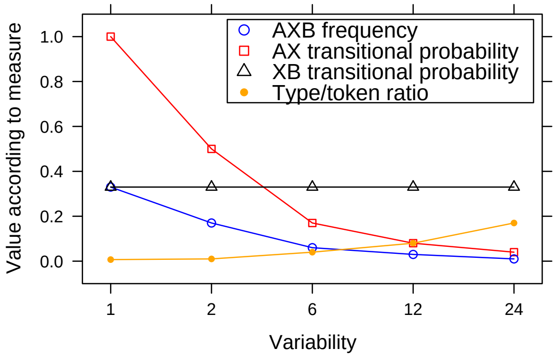

Figure 2: Statistical cues do not explain variability effects.

better in some conditions than in others? Onnis et al. (2003) argued that learners attempt to seek invariance in the input. When variability is high, dependencies stand out as invariant against the fillers and get noticed. When there is no variation in fillers, fillers stand out against the variable frames and attention focuses on dependencies in these frames. In conditions of medium variability, neither frames nor fillers attract special attention leading to poorer performance. Hence, the authors explained the U-shaped behavior by means of an attentional mechanism that tries to detect figure-ground relationships in the input. While this explanation works to account for the big picture, some more fine-grained aspects of the data are left unexplained. For low variability (2), for example, the difference in the number of frames and fillers is smaller than for variability 6, and yet performance was better. Secondly, no significant difference between variabilities 12 and 24 was found, although the invariance account predicts that more variation in fillers should facilitate learning here. Thus the postulated attentional mechanism may not fully explain behavioral differences between conditions.

In this paper we present a connectionist model that replicates human performance on the dependency learning (and generalization) task. The model suggests an alternative, similarity-based explanation of the variability effect that does not involve the role of attention. Differences in model behavior resulted from the nature of information states induced by the input stream: variability in the fillers exerted two opposing forces which conspired to produce the U-shaped pattern in a single-route mechanism. The model allowed us to make precise, quantitative predictions when the number of frames and the dependency distance were increased. We conclude with a discussion of our approach.

The liquid-state framework

Liquid-state machines (LSM for short) are recurrent neural networks which are modelled on the information processing characteristics of the cerebellum (Maass et al., 2002). Their defining characteristic is a sparsely and randomly connected reservoir of neuron-like units (liquid) which turns a time-varying input signal into a spatio-temporal pattern of activations (Figure 3). Recurrence in the liquid equips the model with a working memory of past inputs that is degrading over time (similar to the context layer of an SRN, see Elman, 1990).

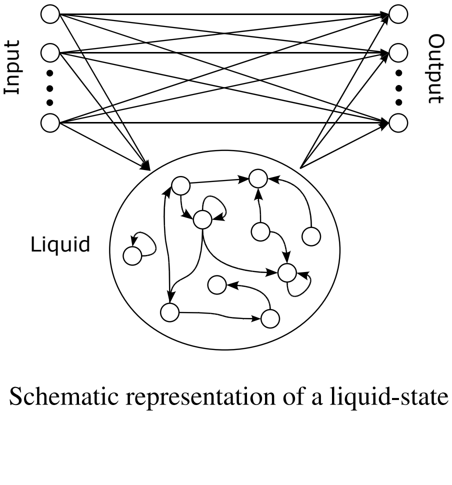

Figure 3: Schematic representation of a liquid-state machine.

A *liquid state* is a vector of liquid-size in which each position corresponds to the activation value of one unit in the reservoir. During processing the liquid state is updated according to the formula

$$z(t+1) = \sigma(\mathbf{w}_{\text{liq}} z(t) + \mathbf{w}_{\text{in}} x) \tag{1}$$

where $z(t)$ is the liquid state at time t, $\mathbf{w}_{\text{liq}}$ is the connection matrix of the liquid, $\mathbf{w}_{\text{in}}$ is the connection matrix from the input units to the liquid, x is the current input, and σ is the activation function of units in the liquid (in our implementation tanh). The liquid consisted of 60 units and connectivity was set to 10%. To make the liquid *state-forgetting* (Jaeger, 2001), the spectral radius of $\mathbf{w}_{\text{liq}}$ was clamped to 0.9 and the connection matrix was scaled accordingly.

Input to the model was encoded using ten units. Each symbol in the language was represented by five randomly chosen units of the input layer. These were switched to 1, the rest to 0. The same distributed encoding was used to represent target symbols at the output layer. The model's output was decoded by mapping the five most active units to 1, the others to 0. To predict a target element correctly, the decoded output pattern had to match the target's encoding exactly. Thus, the model was not guaranteed to predict elements of the appropriate class (A, X or B) in each position. It had to learn word classes and positional information from the input.

A sequence of inputs to an LSM induces a diverse range of nonlinear dynamics in the liquid. In order to compute with an LSM, a set of linear output units is calibrated to map the internal dynamics to a stable, desired output. Calibration (or training) can be achieved by adjusting $\mathbf{w}_{\text{out}}$, the weights from the input and liquid to the output layer, using multiple linear regression

$$\mathbf{w}_{\text{out}} = (S^t S)^{-1} S^t T \tag{2}$$

where S is the collection of internal states during the presentation of an input sequence (and S^t its transpose), and T is the matrix of targets that the model is intended to produce. In other words, to train an LSM an input sequence is passed through the liquid *once* and subsequently the read-out weights are adapted such that the sum of squared residuals is minimized at the output layer. All other weights in the model, most importantly the liquid itself, remain unchanged. Regression training boils down to matrix inversion which is cheap to compute. To avoid singularity a small amount of Gaussian noise ($\mu = 0$, $\sigma^2 = 0.001$) was added to each bit of an input pattern. This proved sufficient to ensure that the inverse (of $S^t S$) always existed.

LSMs have previously been used in natural language processing tasks, e.g., in speech recognition (Triefenbach et al., 2011), grammar learning (Tong et al., 2007; Frank & Čerňanský, 2008) and reading time prediction (Frank & Bod, 2011). We present the first application of these models that aims at explaining a particular psycholinguistic phenomenon.

Learning and generalization

The model was trained on artificial languages similar to those used in the AGL studies—three frames $A..B$ interspersed with X elements drawn from sets of various sizes (1, 2, 6, 12 and 24). For each level of variability, all grammatical strings were generated, they were concatenated in randomized order, and these blocks were repeatedly presented to the model for a total of 432 strings. To mimic the 750ms pause between items in the human experiments, each string AXB was followed by an end-of-sentence marker P. As in the AGL studies, the model received the training set as one continuous input stream without being reset between items. The test procedure differed from the human task of judging the grammaticality of strings. The model rather had to predict the next element in a test sequence, and was evaluated on how well it predicted the dependent elements (Bs). The test set consisted of all string types that the model had encountered in training. Individual differences in human subjects were simulated by randomizing the distributed input representations between model runs. This also minimized the risk of observing behavior that was an artefact of a particular encoding. Results were averaged over 12 model subjects as in the AGL experiments.

After training, the model was 'well-behaved' in that it had learned the transitional probabilities for both A and X elements in each variability condition. This indicates that the training procedure was adequate to track adjacency information in the input. Predicting the B elements in trained items, the model displayed a U-shaped curve which was qualitatively similar to human subjects (Figure 4), although performance was substantially better in high variability conditions (12 and 24) and worse for variability 6. Overall, though, the model matched the human data on how dependency learning is influenced by filler variability quite well.

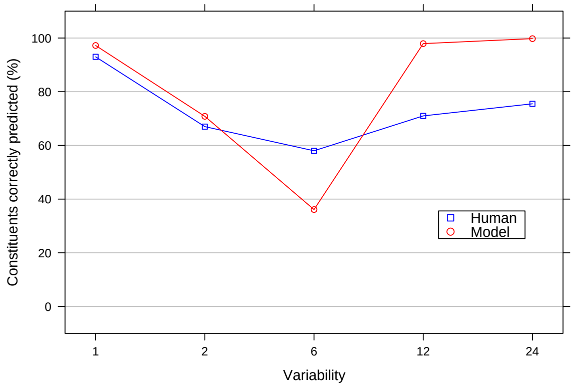

Figure 4: The model showed U-shaped performance similar to humans when tested on trained strings AXB.

Onnis et al. (2004) investigated whether there was also a variability effect when subjects had to generalize to novel items. Tested strings now contained fillers X that did not occur in training. The model's ability to generalize was measured in a similar way, by testing on 6 strings composed of familiar frames with novel X elements (e.g., $A_2X_{31}B_2$). On this task, the model again closely matched human behavior for zero and high variability, although it did not generalize nearly as well as humans for medium variability (Figure 5). In both test conditions—trained and novel fillers—the model amplified human differences between variabilities but showed considerable qualitative similarities with the data.

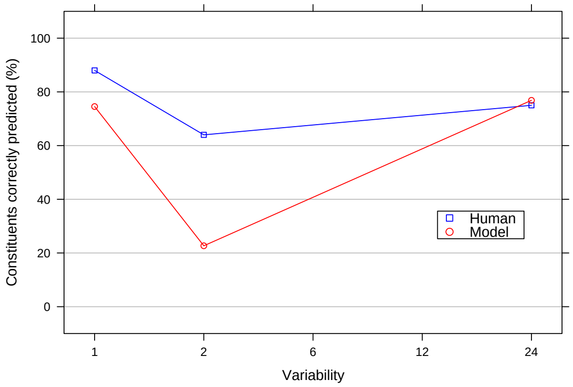

Figure 5: U-shaped performance for strings with novel fillers X (human data not available for variability 6 and 12).

Robustness

Neural network models are often sensitive to small changes in parameters, initialization, and training conditions. We found that the LSM was very consistent in its behavior. Changes in the language encoding, liquid size, percentage connectivity, spectral radius, and amount of noise did not essentially alter the model's U-shaped behavior, although, of course, performance was closer to the human data in some settings than in others. In similar vein, varying the total number of cycles through randomized blocks of stimuli or the time-scale of updating the liquid did not lead to a qualitative change in behavior. It was almost impossible to erase the characteristic differences except when the liquid was so small that the model did not learn dependencies above 10% in any condition. This suggests that the LSM had a strong architectural propensity to differentially respond to relevant information depending on the amount of variation in the input.

Model analysis

The critical information that was used to train the LSM was contained in the states of the liquid while the input sequence was passed through it. If inputs were sufficiently similar they caused the liquid to assume similar states and eventually got mapped to the same output; if inputs were sufficiently dissimilar the liquid separated them at the output. To analyze the model's behavior, internal states were recorded during the input phase, a principal components analysis was conducted, and the liquid was visualized by projection into a two-dimensional principal subspace. After presentation of an X element, the liquid entered a state from which a dependent element had to be predicted (*B*-state for short). For zero variability, variation in *B*-states derived entirely from distinct *A* elements whereas in the other conditions also differences in *X* elements added variation. As variability increased from 1 to 24, the regions from which identical *B* elements had to be predicted increased in size because distinct *X* elements sent the liquid into distinct states. This steady increase in state dispersion was measured as the average Euclidean distance of a *B*-state region from its centroid. At the same time, increasing variation in *X* elements provided more and more distinct data points in each such region. Thus, variability had two opposite effects on the information states that the model used to predict *B* elements. *B*-state regions that mapped to identical dependencies grew larger and simultaneously became filled more densely with relevant training data (see Figure 6). When combined, these two forces—dispersion and density—could explain U-shaped performance.

Since trained *B*-states resulted from a continuous stream of input sentences, and tested *B*-states from presenting a single test sentence, the former always deviated slightly from the latter. In testing, the model could correctly predict a dependent element if the corresponding *B*-state was sufficiently close to a *B*-state that the model had assumed in training. For zero variability, *B*-states clustered in a small region of state space that contained many data points because there were 144 cycles through the training set in this condition (Figure 6, left). *B*-states in testing mostly fell into this region (and the model made a correct prediction) due to the lack of variation in the *X* element. For high variability, *B*-states spanned a larger region of state space, with distinct data points deriving from all trained items with a different *X* element. As a consequence, the training algorithm adapted the entire region of state space to map onto the same *B* element (Figure 6, right). This made the model highly robust for variability 24, especially in the generalization task with novel *X* elements. When variability was medium (Figure 6, center), states that mapped to the same *B* were scattered in isolated clusters across state space (one for each distinct *X* element) and these clusters each contained less data points than in the zero variability condition. When *B*-states in testing fell outside these regions, the model could not interpolate the dependent element as in the high variability condition, and hence accuracy was lower.

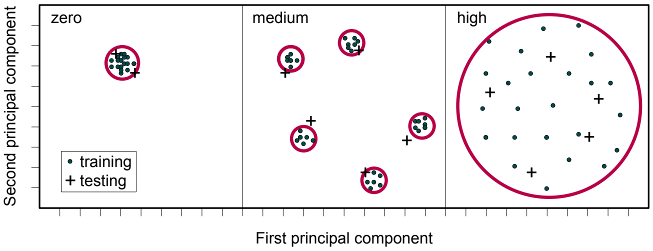

Figure 6: *B*-states in training (dots) and testing (crosses) for three levels of variability; simplified depiction.

To verify that this was the correct analysis, we derived three predictions that were then tested experimentally. For zero variability, prediction accuracy should drop when there is only one cycle through the training set. Now there is only a single data point in the circular region of Figure 6 (left) to which the model is adapted in training, and *B*-states induced in testing are not entirely congruent. High variability conditions, on the other hand, should be less affected by the number of cycles. This was confirmed in that accuracy in the zero variability condition dropped to 0% and remained above 90% for variability 24. Secondly, imposing a large amount of noise on the liquid's internal states should increase the area of *B*-state regions in training and thus make the model more fault-tolerant. In conditions of medium variability, this should improve prediction success, and indeed the model reached almost 100% accuracy on trained items for variability 6. And third, the model should also achieve very high accuracy when variability is increased to 48 because the critical *B*-state region should become even more densely filled with training data than for variability 24. This prediction was confirmed as well, the model reached above 90% accuracy on both trained and novel items when variability was increased to 48.

Apart from these factors, the choice of learning algorithm can have a strong influence on neural network behavior. Thus, it is possible that the reported results were mainly due to regression training. To determine the role that the training regime played in creating the observed behavior, we compared the LSM with a feed-forward network. This network had the liquid replaced by a non-recurrent hidden layer but was identical otherwise. Without recurrence, the model did not implement a working memory and hence could not predict dependent elements above chance. Nonetheless, if regression training played a crucial role we would also expect to witness similar U-shaped accuracy in the feed-forward network (relative to chance level performance which was identical in all conditions). We found that this was not the case; model performance peaked for variability 6, and was lowest for variability 24. For novel items, there was a steady decline in accuracy from zero to high variability. This control experiment suggests that the effect of differences in filler variability on dependency learning was caused by the properties of the liquid and not by the training algorithm that was used.

Novel predictions

Frames in natural language can be more diverse than in the AGL experiments, and dependencies can be separated by more than one word. We therefore tested the model in conditions of increased frame variability and dependency distance.

Increased frame variability

According to the standard explanation of the variability effect, learners seek to identify invariance in the input (Onnis et al., 2003, 2004). When variability in *X* is high, the frames *A*..*B* stand out as invariant against the *X* elements. When variability in *X* is zero, the focus shifts on the variation in frame dependencies. In conditions of medium variability, the number of frames and fillers is similar, which makes it difficult for the learner to detect dependencies and this results in lower performance. To assess this account in the model, the number of $A..B$ frames in the language was doubled. If the invariance account is correct, we should observe improved performance for variabilities 1 and 2 because the frame-to-filler ratio increases. For variability 6, we should observe a drop in performance because having the same number of different frames and fillers in the input should mask frame invariance. For high variability 12 and 24, we should also observe a drop in performance because the difference in the number of frames and fillers is less distinct. Figure 7 shows the model's learning behavior for six frames compared to the results of Figure 4 for three frames. For high variability (12 and 24) there was no performance difference and for medium variability (6) performance improved considerably. In the zero and low variability conditions (1 and 2), the model performed worse than before. Thus, the U-shaped pattern persisted when dependencies in the language were more complex and the results suggest that the behavior of the LSM was not in accordance with the predictions of the invariance account.

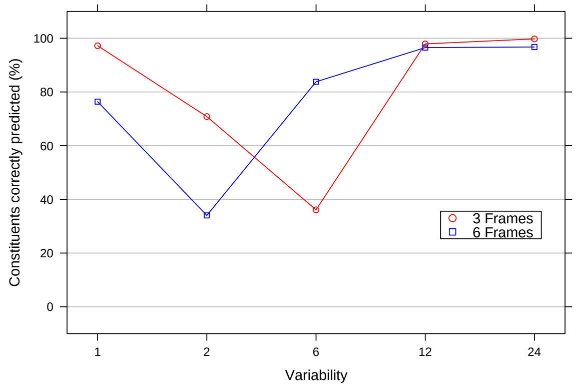

Figure 7: Performance increased for medium variability when the number of frames was doubled.

Increased dependency distance

In a third experiment, the model was used to investigate performance for increased distance between nonadjacent elements. The input language consisted of 'sentences' $AXYB$, where the filler chunks XY were again drawn from sets of cardinalities 1, 2, 6, 12 or 24. The invariance account does not make predictions for increased distance since it does not specify the role of working memory in learning nonlocal dependencies. In the model, U-shaped behavior persisted for trained items, and to some extent also for novel filler chunks (Figure 8). Compared to Figure 5, however, increasing the distance led to a breakdown in generalization. A novel combination of two fillers caused the model to enter regions of state space that could not reliably be mapped to dependent targets by the read-out units. As in the previous experiment of increased frame variation, the most pronounced difference occurred for variability 6. In both these experiments, there was more variation in B-states resulting from either more frames or fillers in the input language. This variation helped the model to better predict dependencies for variability 6 in a way similar to the effect of adding noise to the liquid in the control condition described in the analysis section.

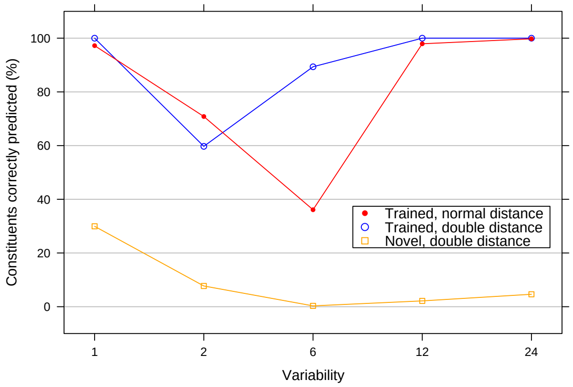

Figure 8: Learning and generalization for increased dependency distance.

General discussion

Several recent AGL studies have shown that the learning of nonadjacent dependencies is modulated by the amount of variation in the filler elements (Gómez, 2002; Gómez & Maye, 2005; Onnis et al., 2003, 2004). The U-shaped pattern found in these studies can not easily be explained by recourse to distributional properties of the language input and is difficult to reconcile with many findings indicating that the human language system is remarkably sensitive to transitional probabilities (e.g., Saffran et al., 1996). In particular, these results pose a challenge for statistical models of language learning that exploit adjacency and frequency information.

To account for this data we used a liquid-state model which is a sparsely connected, recurrent network that computes over transient states. LSMs implement a working memory to detect temporal contingencies and can be trained efficiently by linear regression. This allowed us to study the model's behavior after exposure to the same small number of training items as in the AGL studies. The LSM was trained off-line after the entire input sequence had been presented. This procedure may not be not faithful to the human experiments where implicit expectations about upcoming words might be formed during the input phase already. However, it was shown that the training regime was not critically responsible for the observed U-shaped behavior. It remains to be tested whether the model displays similar behavior when the read-out weights are adjusted incrementally (e.g., using perceptron learning).

The liquid-state approach provides a generic neuro-computational framework for sequential processing and cognitive modelling more broadly. The liquid is general purpose and can be used to model an indefinite number of cognitive tasks (even in parallel). Connectivity in the liquid is not altered during learning and hence these models make very modest assumptions about the nature of mental representations. Inputs which are sufficiently distinct are separated by the liquid, inputs which are sufficiently similar are mapped to similar output. In such a system, differential behavior results from the input stream filtered through the architecture of the model, rather than the observable symbolic properties of the input itself. That is to say, variability in the input generates statistically relevant information in the liquid—such as the density and dispersion of information states—that is not measurable in the input stream in terms of transitional probabilities, *N*-gram frequency or the type-token ratio. The explanation we propose for the variability effect is based on these properties of information states. It is a hallmark of neural network models that they represent inputs as a graded pattern of activation distributed over a set of units. In the LSM, differences and similarities between such patterns were picked up by the regression used to calibrate the read-out weights. This enabled the model to categorize novel stimuli based on their representational similarity with trained items. When variability in fillers was zero, representations of test stimuli were highly similar to those of trained items because they fell into a small region of state space that was densely populated by training data. When variability was high, a large region of state space was adapted to map to the same dependency in training which again caused high similarity between trained and tested items. For medium variability, similarity was lower because the state space got partitioned into smaller, separate regions. When representations of test items fell outside these regions the model produced errors in predicting dependencies.

This similarity-based account differs from the invariance account proposed in Onnis et al. (2003). Whereas the latter argues that differential learning can be explained by a mechanism of attentional shift that seeks to find stable patterns in a noisy stream, the account proposed here is based on similarities between information states in the learner's working memory which are induced by training and test stimuli. The model suggests that this might be an alternative, more parsimonious explanation of the variability effect. Both accounts, however, are not mutually exclusive although some of the model predictions were not in line with the invariance account. Attention as well as representations in working memory might play a role in learning nonadjacent dependencies, especially in the implicit learning paradigm in which all of the data on the variability effect have been gathered.

We also used the LSM to obtain novel predictions for conditions in which there were more frames in the language (six instead of three) and a larger distance between dependencies (two fillers instead of one). It was found that the model displayed a similar U-shaped pattern, but shifted towards lower variabilities. The most pronounced difference occurred for medium variability where the model's performance improved significantly compared to the standard condition (three frames, one filler). These precise, quantitative predictions suggest a straightforward test of whether processing in the model adequately captures the implicit learning of nonadjacent dependencies in humans. In future work we therefore intend to assess this similarity-based, liquid-state model account in AGL behavioral experiments.

Acknowledgments

Thanks to Stefan Frank and Morten Christiansen for helpful comments and discussion.

References

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

Frank, S., & Bod, R. (2011). Insensitivity of the human sentence-processing system to hierarchical structure. *Psychological Science*. (In press)

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

Gómez, R. (2002). Variability and detection of invariant structure. *Psychological Science*, *13*, 601–617.

Gómez, R., & Maye, J. (2005). The developmental trajectory of nonadjacent dependency learning. *Infancy*, *7*, 183–206.

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

Leibbrandt, R., & Powers, D. (2010). Frequent frames as cues to part-of-speech in Dutch: Why filler frequency matters. In *Proceedings of the 32nd Annual Conference of the Cognitive Science Society* (pp. 2680–2686). Austin, TX: Cognitive Science Society.

Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, *14*, 2531–2560.

Mintz, T. (2003). Frequent frames as a cue for grammatical categories in child directed speech. *Cognition*, *90*, 91–117.

Onnis, L., Christiansen, M., Chater, N., & Gómez, R. (2003). Reduction of uncertainty in human sequential learning: Evidence from artificial grammar learning. In *Proceedings of the 25th Annual Conference of the Cognitive Science Society* (pp. 887–891). Mahwah, NJ: Lawrence Erlbaum.

Onnis, L., Monaghan, P., Christiansen, M., & Chater, N. (2004). Variability is the spice of learning, and a crucial ingredient for detecting and generalising nonadjacent dependencies. In *Proceedings of the 26th Annual Conference of the Cognitive Science Society.* Mahwah, NJ: Lawrence Erlbaum.

Saffran, J., Aslin, R., & Newport, E. (1996). Statistical learning by 8-month-old infants. *Science*, *274*, 1926–1928.

Tong, M., Bickett, A., Christiansen, E., & Cottrell, G. (2007). Learning grammatical structure with echo state networks. *Neural Networks*, *20*, 424–432.

Triefenbach, F., Jalalvand, A., Schrauwen, B., & Martens, J.-P. (2011). Phoneme recognition with large hierarchical reservoirs. In *Neural Information Processing Systems (NIPS 2010).* (In press)