Strong systematicity in sentence processing by simple recurrent networks

Philémon Brakel (pbpop3@gmail.com)

Stefan L. Frank (S.L.Frank@uva.nl)
Institute for Logic, Language, and Computation, University of Amsterdam
Science Park 904, 1098 XH Amsterdam, The Netherlands

Abstract

Providing explanations of language comprehension requires models that describe language processing and display strong systematicity. Although various extensions of connectionist models have been suggested in order to account for this phenomenon, we found that even a simple recurrent network that had been trained in a way that can be considered 'standard', could display strong systematicity. This ability was found to result from informative word representations that developed in the network.

Keywords: Simple Recurrent Network; Systematicity; Sentence Processing; Word Representations; Hierarchical Cluster Analysis.

Introduction

Models that want to account for natural language sentence processing also have to explain how humans can produce a large number of novel sentences after being exposed to a finite number of utterances. This ability of a model to combine words in new ways is often referred to as systematicity[1]. Fodor and Pylyshyn (1988) argued that connectionist networks are intrinsically unable to display systematicity unless they implement a symbol system.

To display systematicity, a network needs to generalize. A model generalizes successfully if it learns regularities from a set of training data that allow it to increase its performance on processing data that has not been encountered before. According to Hadley (1994), one can divide the extent to which this generalization is necessary for sentence processing into weak and strong systematicity. Weak systematicity is the ability to process sentences that are different from the sentences the system was trained on but have words occurring in the same positions as during training. An example of this is when someone learns from sentences 1a and 1b to understand sentence 1c as well.

(1) a. Dogs follow men
b. Women follow dogs
c. Women follow men

Strong systematicity requires more abstraction and is defined as the ability to process test sentences with words occurring in different positions than in the training sentences. An example of strong systematicity is when someone is exposed to sentences 2a and 2b and learns from those how to process sentence 2c.

(2) a. Dogs follow men
b. Women follow dogs
c. Men follow women

Moreover, the model should also be able to process sentences that contain embedded clauses with words occurring in novel positions.

Hadley (1994) argues that humans display strong systematicity. He continues that connectionist models have been shown to display weak systematicity but that it has yet to be shown that they display strong systematicity as well. Strong systematicity is the type of systematicity that we will test neural networks on in this paper.

It has been argued that merely demonstrating systematicity in a neural network does not suffice to disprove the claims by Fodor and Pylyshyn (1988). Rather, it needs to be shown that connectionist systematicity (without merely implementing a symbol system) is not only possible, but necessarily follows from the network's architecture (Aizawa, 1997). As Fodor and McLaughlin (1990) put it, it is a law (and not just an accident) that cognition is systematic.

Although this law-requirement has been questioned (e.g., Hadley, 1997), we do agree that, for demonstrations of connectionist systematicity to be convincing, they should not depend on some arbitrary arrangement of the system or the training regime, nor on architectures or representations that were developed for obtaining systematic behavior. Moreover, systematicity should arise robustly and in a variety of circumstances, rather than showing up in just one or a few instances.

Many demonstrations of connectionist systematicity have not met these standards. For example, Christiansen and Chater (1994) showed some indication of strong systematicity with a simple recurrent network in a single simulation, but without carrying out a thorough quantitative analysis. Additionally, Hadley, Rotaru-Varga, Arnold, and Cardei (2001) showed systematicity with a hebbian competitive network while explicitly adding semantic information to obtain this goal. Bodén (2004) used a specially designed cascaded architecture to obtain systematic behavior with recurrent networks. Recently, Frank and Čerňanský (2008) showed sytematicity in a recurrent neural network but only when supplying it with pre-arranged input representations. Again, this extension was included just to obtain systematic behavior. Besides the research of Christiansen and Chater (1994), all these earlier attempts utilised network architectures or representations that had been recruited for the purpose of showing systematicity. By doing so, little evidence is provided for a more general display of generalization that is robust and independent of the

[1] Not to be confused with the possibility of an infinite number of grammatical sentences in a language which would be generativity.

specific circumstances.

In the current paper, we show that even a standard neural network architecture for sentence processing can display strong systematicity in a large number of test inputs, without requiring a special training regime or any information that is not available from the training data. We argue that it does so without implementing a symbol system.

Simple recurrent networks

The best known neural network architecture that can process sequential information is the simple recurrent network (Elman, 1990) (SRN). SRNs can learn various types of dependencies in the presented data and represent quite complex automata. We chose to use this architecture because it is very common in language modeling and able to update both the way it represents data and the way it uses data to predict new words. This is unlike the echo state network Frank and Čerňanský (2008) used. In echo state networks, only the weights to the output layer are being trained. Therefore, the echo state network does not change the way it represents data and Frank and Čerňanský (2008) had to explicitly construct input representations in advance in order to obtain strong systematicity. By using an SRN in the current study, we will not use any external means to obtain input representations that are based on prior intuitions.

Prediction and systematicity

One way to investigate the ability of SRNs to generalize (and display strong systematicity) is to present sentences to them and train them to predict upcoming words. It is not possible to predict the next word of a sentence with certainty, so the goal of the prediction tasks is to estimate a probability distribution over the words in the lexicon.

We used a probabilistic context-free grammar to generate training and test sentences. After this, we used the prediction task to investigate how well a network does at discovering the underlying generator of the artificial language. In the original grammar of this language, all nouns are treated equally and can appear in all positions that a noun can normally appear in. During training however, a distinction is made among different types of nouns based on gender. This restricts the precise grammatical roles the nouns can take in a sentence. The test sentences contain the same nouns but in grammatical roles in which they never appeared during training. The only way to still perform well on word prediction, is to discover that there is a general category of nouns that can appear in all those positions regardless of gender. The model has to display strong systematicity by learning that the words can also form sentences by appearing in different positions than the ones they appeared in during training.

Measuring systematicity

As a criterion for strong systematicity, Frank and Čerňanský (2008) argued that a model should outperform the best nth order Markov model that has been constructed from the training data. In a Markov model, the probability of each possible upcoming word depends on a specific number of preceding words in the current sentence. Formally this means that, according to an nth order Markov model, the probability that a certain word i will follow a sequence that is n words long is defined as

$$Pr(i|w_{t-n},\dots,w_{t-1}) = \frac{N(w_{t-n},\dots,w_{t-1},i)}{N(w_{t-n},\dots,w_{t-1})}, \qquad (1)$$

where $N(\cdot)$ gives the number of occurrences of its argument in the training data. This definition entails that a 0th order Markov model (unigram) is just based on the frequency of the word itself. A 1st order Markov model (bigram) depends just on the relative frequency of the upcoming word and the number of times that it was seen given the current word in the sequence. Larger values for n do not always result in better performance.

Whenever a Markov model processes a combination of words that did not appear in the training data, it can only base its prediction on the last words that form a combination that *did* occur in the training data and on the frequencies of the different words. This makes it impossible for the Markov model to use words that appear earlier in the sentence. For this reason Markov models don't take into account dependencies that are based on discontiguous word co-occurrences. The performance of the best Markov model can be seen as the best a well-trained but non-systematic model can do with the current data. This implies that obtaining a higher performance than the best Markov model requires the ability to display systematicity and learn aspects of the data that are useful for dealing with new unobserved combinations.

The bigrams, trigrams or greatest nth order Markov models might not always be the Markov models with the highest performance. For this reason, comparing with the performance of just one of these Markov models might result in classifying other models as systematic too easily if another Markov model with better performance exists. To classify a model as systematic, it is not sufficient to just compare it with bigrams and trigrams or with the greatest nth order Markov model as done by Tong, Bickett, Christiansen, and Cottrell (2007) and Frank (2006), respectively. Systematicity (and particularly strong systematicity), should be impossible to achieve for *any* nth order Markov model. The Markov model that serves as a baseline should be the one that achieves the best performance for the current sequence. This requires calculating all nth order Markov models for the current sequence until the beginning of the sentence has been reached and deciding which n-gram performs best afterwards.

Normally one could not use the *best* nth order Markov models to find regularities in sequential data as described above because it is impossible to select the best performing model without already knowing the true underlying probabilities. However, in the context of the current research, this information is available. The best Markov model should not be seen as a language model but as a baseline model that has been made very strict by providing it with prior knowledge.

Table 1: The Probabilistic grammar that underlies the artificial language used in the experiment. Variable r denotes grammatical role. When the probabilities of different productions are not equal they are indicated within parentheses.

S	→	NP$_{subj}$ V NP$_{obj}$ *[end]*
NP$_r$	→	N$_r$ (.7) | N$_r$ SRC (.06) | N$_r$ ORC (.09) | N$_r$ PP$_r$ (.15)
SRC	→	*that* V NP$_{obj}$
ORC	→	*that* NP$_{subj}$ V
PP$_r$	→	*from* NP$_r$ | *with* NP$_r$
N$_r$	→	N$_{fem}$ | N$_{male}$ | N$_{anim}$
N$_{fem}$	→	*women* | *girls* | *sisters*
N$_{male}$	→	*men* | *boys* | *brothers*
N$_{anim}$	→	*bats* | *giraffes* | *elephants* | *dogs* | *cats* | *mice*
V	→	*chase* | *see* | *swing* | *love* | *avoid* | *follow* | *hate* | *hit* | *eat* | *like*

Even with the availability of prior knowledge, the Markov models might not perform well because of their discrete way of handling unseen information. Normally, the Markov models would assign a probability of zero to word combinations that have not been encountered yet. This could allow a noisier model to outperform the Markov models in these cases without truly displaying systematicity.

For these reasons, and unlike Frank and Čerňanský (2008), we will use the *smoothed* best nth order Markov models as a baseline. We applied Laplace smoothing where the frequency of unseen combinations was assumed to be one instead of zero. Earlier tests showed that the application of smoothing resulted in slightly better performance for the best Markov models so it makes the baseline even more strict.

Experiment

Language

To generate training and test sentences, the same probabilistic context-free grammar was used as in the experiment by Frank and Čerňanský (2008). This language has 26 different words. From these words, 12 are nouns, 10 are transitive verbs and 2 are prepositions. There are also a relative clause marker and an end-of-sentence marker. Its production rules (see Table 1) allowed for complex sentences, containing for example center-embedded clauses.

Training sentences

Training sentences were created following the rules of the artificial grammar in Table 1 but with some extra restrictions so that only a subset of all possible sentences was generated. This was done by replacing the N$_r$ production rule with two rules stating that the subject of a sentence always had to be either a female noun or an animal noun and the object always a male noun or an animal noun (see Table 2). These rules are not part of the original underlying grammar. The set of training sentences would therefore, never completely converge towards the underlying distribution as specified by the grammar in Table 1.

Table 2: Production rules replacing N$_r$ that were used for generating the training sentences.

N$_{subj}$	→	N$_{fem}$ | N$_{anim}$
N$_{obj}$	→	N$_{male}$ | N$_{anim}$

Table 3: The four types of sentences that were used for testing on systematicity.

SRC1:	N$_{male}$	*that*	V	N$_{fem}$	V	N$_{fem}$	*[end]*
SRC2:	N$_{male}$	V	N$_{fem}$	*that*	V	N$_{fem}$	*[end]*
ORC1:	N$_{male}$	*that*	N$_{male}$	V	V	N$_{fem}$	*[end]*
ORC2:	N$_{male}$	V	N$_{fem}$	*that*	N$_{male}$	V	*[end]*

Test sentences

The test sentences had male nouns in the subject positions, female nouns in the object positions (see Table 3) and contained no animal nouns. They could not have been generated by the more restricted grammar that generated the training sentences and strong systematicity is required to process them correctly. Note for example, that due to this reversal of the grammatical roles for the male and female nouns, the first word of the test sentences is always a male noun. This is in contrast with the training sentences that always started with either animal or female nouns.

The four types of test sentences are shown in Table 3 and are labeled with the abbreviations SRC1, SRC2, ORC1 and ORC2 that refer to two types of subject relative clauses (SRC) and object relative clauses (ORC). All these types of sentences contained three nouns that had to be either male or female (chosen from three possible variants) and two verbs (chosen from a set of ten). The total number of systematicity test sentences equaled 10800.

Network architecture

We used the same architecture for the SRNs as described in Elman (1990) but with different numbers of hidden units. Words get presented at the input layer from which the weighted activation is fed forward to the recurrent layer. The recurrent layer's activation is computed from a projection of the input activation, a bias vector and a transformation of its activation at the previous time-step. This last component allows the network to learn sequential information. The final activation of the recurrent layer is calculated with the logistic sigmoid activation function which is defined as

$$a_{rec,i} = f(x_i) = \frac{1}{1+e^{-x_i}}, \qquad (2)$$

where x_i is the total activation arriving at unit i. The output layer receives input from the recurrent layer and its own bias vector after which the *softmax* activation function is applied. This function is defined as

$$a_{out,i} = f_{out}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}. \qquad (3)$$

The softmax function results in positive real values for the activation vector of the output layer that sum to one. This makes it possible to interpret them as probabilities. More precisely, the activation vectors per layer are given by

$$\begin{aligned} \mathbf{a}_{\text{rec}}(t) &= \mathbf{f}(\mathbf{W}_{\text{in}}\mathbf{a}_{\text{in}}(t) + \mathbf{W}_{\text{rec}}\mathbf{a}_{\text{rec}}(t-1) + \mathbf{b}_{\text{rec}}) \\ \mathbf{a}_{\text{out}}(t) &= \mathbf{f}_{\text{out}}(\mathbf{W}_{\text{out}}\mathbf{a}_{\text{rec}}(t) + \mathbf{b}_{\text{out}}), \qquad (4) \end{aligned}$$

where $\mathbf{a}_{\text{in}}(t)$ is a 26 dimensional vector that contains zeros in all of its elements except for a 1 in the element that corresponds to the input word at time t. Respectively, the vectors $\mathbf{a}_{\text{rec}}(t)$ and $\mathbf{a}_{\text{out}}(t)$ contain the activation values of the recurrent and output layers at time t, $\mathbf{W}_{\text{in}}$ is the matrix containing the input weights and $\mathbf{W}_{\text{rec}}$ and $\mathbf{W}_{\text{out}}$ are similarly the matrices containing respectively the recurrent and output weights. The vectors $\mathbf{b}_{\text{rec}}$ and $\mathbf{b}_{\text{out}}$ contain the bias vectors that are used for respectively the recurrent layer and the output layer.

Training

Backpropagation and stochastic gradient descent were used to train the network by minimizing the cross-entropy error-function instead of the well-known sum-squared-error that was being minimized in Frank and Čerňanský (2008). Minimizing the cross-entropy for a multi-class classification problem is equivalent to maximizing a log-likelihood function that is based on the categorical distribution. Simard, Steinkraus, and Platt (2003) found faster convergence and better generalization properties by classification networks that were trained by minimizing cross-entropy instead of the sum-squared-error.

A group of ten networks with 41 hidden units (only differing in the random[2] values of their initial weight matrices) was trained for 10 epochs on 25000 sentences with an average length of 5.9 words. The training sentences were presented word by word and the networks were trained to associate each word with the next word in the sequence. A learning rate of 0.1 was used and decreased with 0.01 after each training epoch.

Results and discussion

To obtain performance scores for the SRNs and the Markov models, the cosines of the angles between the vectors with the probability estimations of these models and the true probability distributions derived from the underlying probabilistic context-free grammar were calculated. This distance measure is close to 1 if the vectors are highly similar and close to 0 if they are nearly perpendicular.

Table 4: Average performance scores on the training and test sentences for the SRNs and the Markov models.

Data set	SRN			Markov
	best	worst	averaged	best
Train set	.964	.950	.959	.944
Test set	.927	.856	.906	.792

Comparing the models

As Table 4 shows, the average performance of the ten SRNs on the test sentences was substantially higher than that of the smoothed best Markov models. Even the worst performing network outperformed the Markov models. The average performance on the training data was also investigated but since the training sentences were generated at random, these results are not directly comparable.

As Fig. 1 shows, the performance patterns per sentence type of the SRNs and the smoothed best Markov models are quite different. Wilcoxon matched-pairs signed-rank tests showed that these differences were highly significant (with $p < .5 \cdot 10^{-5}$ for all tests) at each position of the test sentences.

Especially when the Markov models scored low, the networks did quite well. The SRNs reached a near perfect performance on the first word of every sentence, while the Markov models had trouble with it. The first word was a male noun and these never occurred in the first position of a sentence during training. To predict the next word, the Markov models could only rely on bigrams (or unigrams) that look at the frequencies of words following a male noun in the object position. In the training data, a male noun in the object position preceded an end-of-sentence marker. Predicting the end-of-sentence marker is clearly incorrect when the first word of the sentence was presented. The networks seem unaffected by this problem. Their predictions seem independent of the gender of the first word but dependent on its grammatical role.

This performance pattern can also be seen in for example the fifth word of the ORC2 sentences. The Markov models do badly at predicting the verb that follows this male noun. In the training data, verbs mainly followed female nouns as those were always subjects. The SRNs however, reach a near perfect performance on this word. Similar results were found for the sixth word of the SRC1 sentences, as well as for several other points in test sentences.

In contrast, the Markov models did a lot better than the networks on the fourth word of the SRC1 sentences but this simply indicates that no systematicity was required to do well on this word. The combination of a verb followed by a female noun never appeared in the training data. Consequently, the best Markov model could only be a bigram or a unigram. Female nouns were always subjects in the training data. It is

[2] The weight matrices were initialised with random values that were uniformly distributed between the interval $[-1, 1]$.

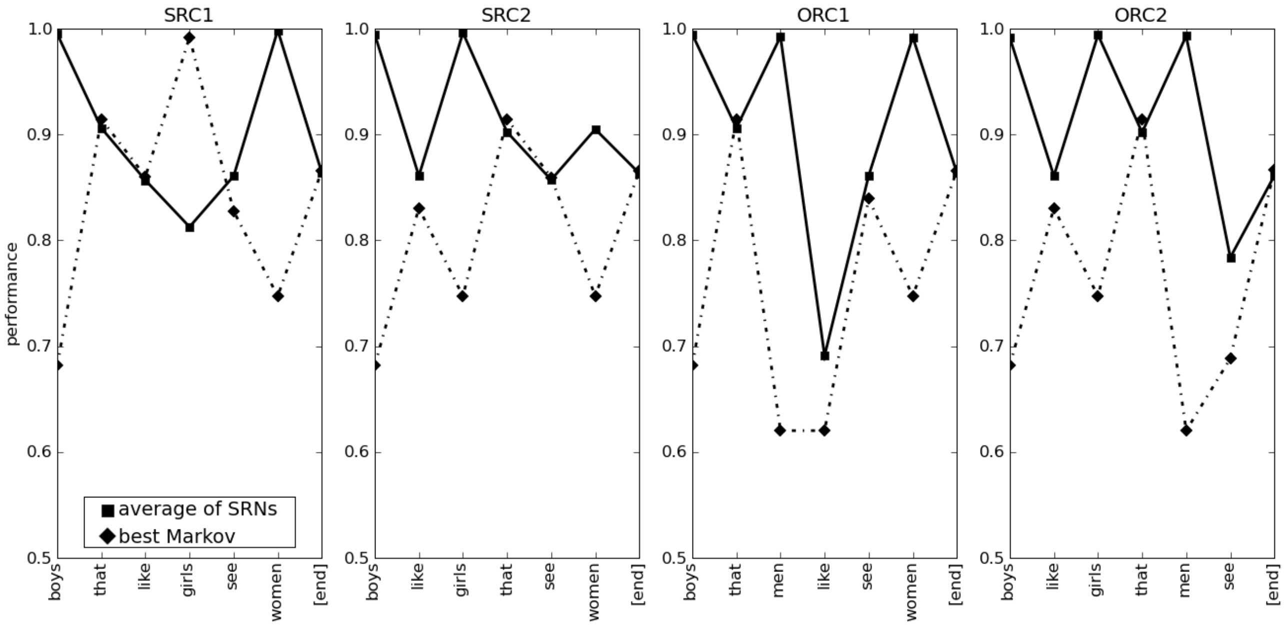

Figure 1: Average performance on each test sentence type, for the SRNs (averaged over all ten networks) and the smoothed best Markov model. Examples of the sentence types are given by the labels on the x-axis

therefore not surprising that the most common bigram predicts that a verb would be likely to follow even though the actual subject of that verb is the male noun that appeared earlier in the sentence. This means that in this case the high performance of the best Markov model results from a lack of systematicity rather than the opposite.

The Markov models performed slightly better at other points but it can be seen that these differences were always very small. Moreover, the performance of the Markov models was relatively high in these cases, reflecting less necessity to display strong systematicity when predicting the next word.

Analysis of the input weights

To get a better understanding of how the recurrent networks might succeed in displaying systematic behavior, we performed a hierarchical cluster analysis (HCA) on the weights that connect the inputs to the recurrent layer. This gives more insight in how the network formed representations of its inputs. We refer to these weight vectors as ‘word representations’ because the localist coding in the inputs causes only a single weight vector at a time to influence the recurrent layer. Note that this is a different notion of word representation than Elman (1990) used.

We took the network with the best average performance for this analysis as it might be most clear from this network how the systematicity came about. Fig. 2 shows the hierarchical clustering of the words based on their distances in the representational space of the input weight vectors. It turned out that the first clustering separates the word *that* and the rest of the words, followed by a distinction between the end-of-sentence marker and the remaining words. After that, the biggest difference is between the nouns and the further remaining words. Within the noun cluster, it seems that further groupings are quite arbitrary and reflect hardly any relation with the gender of the nouns. What is important here is that even though the male and female nouns never appeared in the same grammatical roles, the network was still able to place them in the same general category of nouns together with the animal words. Somehow the network learned to use the grammatical category of the words rather than the exact grammatical roles in which they appeared.

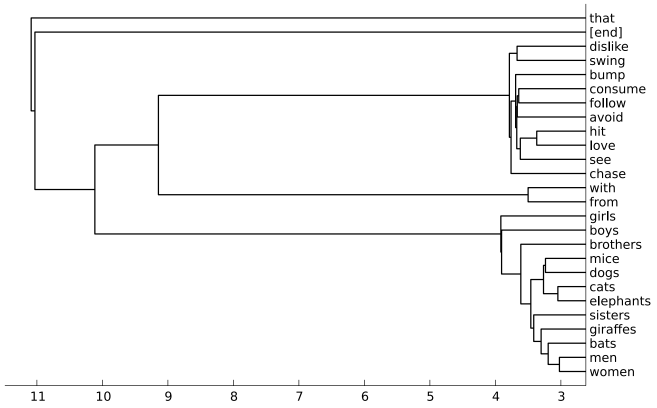

Figure 2: Hierarchical cluster analysis of the input weights of the SRN with the best average performance

Conclusion

SRNs trained to do next word prediction were able to outperform a baseline, that was based on the best smoothed Markov models, on processing sentences that required strong systematicity. Hierarchical cluster analysis showed that SRNs were able to group the words they learned in a way that seemed based on more abstract properties than just their exact grammatical positions.

We conclude that the SRNs displayed strong systematicity and that this provides some evidence against the claim of Fodor and Pylyshyn (1988). We showed that strong systematicity can be displayed by simple recurrent networks that have been trained using a standard methodology. No additional modifications in architecture, training regime or way of representing information were done and no information that was not in the training data was used to train the networks.

According to Fodor and Pylyshyn (1988), combinatorial syntax and semantics are the core properties of a symbol system. Marcus (1998) states that a symbol system is a system that performs operations over variables. A standard SRN is not a combinatorial system and contains no distinction between instantiations of variables and operations over them. We therefore argue that some notions of similarity are sufficient for displaying strong systematicity, without the need for any discrete symbols.

How SRNs generalize

An analysis of the input weights showed that the SRNs were able to form a general category of nouns instead of separate gender based categories. The ability to display systematicity seems to arise from this representational level. Input words that are represented near each other in the representational space result in similar network states. If the female, male and animal nouns are represented close together, they will be treated similarly when predicting the next word.

This similarity plays an important role during training. If animal nouns appear in the same roles as male nouns, they will result in similar states in the network due to the context information that is fed back into the recurrent layer. The network can improve its performance directly by representing them closer together as it follows from the training data that they result in similar context states and require a similar treatment. The same applies to the female nouns and the animal nouns. If the male and female nouns both get represented closer to the animal nouns, it follows (given that the vectors live in a euclidean space) that the distance between the male and female nouns also has to decrease during training.

A language learner who uses frequency information might never be exposed to all positions in which a single noun can occur. However, one could perhaps still learn to use the noun in other positions by representing it closer to other nouns based on *overlap* of contextual properties.

Of course the findings of this experiment are based on one simulation and further research is required. Future studies could investigate the conditions that influence the performance on tasks that require strong systematicity. It would be particularly interesting to see how varying the number of animal nouns might influence performance. If our explanation of how connectionist systematicity arises is correct, increasing the number of animal nouns should improve performance on test sentences.

Acknowledgments

We would like to thank Rens Bod for his support. The research presented here was funded by grant 277-70-006 of the Netherlands Organization for Scientific Research (NWO).

References

Aizawa, K. (1997). Explaining systematicity. *Mind & Language*, *12*, 115–136.

Bodén, M. (2004). Generalization by symbolic abstraction in cascaded recurrent networks. *Neurocomputing*, *57*, 87–104.

Christiansen, M., & Chater, N. O. (1994). Generalization and connectionist language learning. *Mind & Language*, *9*, 273–287.

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

Fodor, J. A., & McLaughlin, B. (1990). Connectionism and the problem of systematicity: Why Smolensky's solution does not work. *Cognition*, *35*, 183–204.

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, *28*, 3–71.

Frank, S. L. (2006). Strong systematicity in sentence processing by an echo state network. In *Proceedings of ICANN 2006, part I* (Vol. 4131, pp. 505–514). Athens, Greece: Springer.

Frank, S. L., & 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.

Hadley, R. F. (1994). Systematicity in connectionist language learning. *Mind & Language*, *9*, 247–272.

Hadley, R. F. (1997). Cognition, systematicity and nomic necessity. *Mind & Language*, *12*, 137–153.

Hadley, R. F., Rotaru-Varga, A., Arnold, D. V., & Cardei, V. C. (2001). Syntactic systematicity arising from semantic predictions in a hebbian-competitive network. *Connection Science*, *13*, 73–94.

Marcus, G. F. (1998). Rethinking eliminative connectionism. *Cognitive Psychology*, *37*, 243–282.

Simard, P., Steinkraus, D., & Platt, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR* (p. 958-962). IEEE Computer Society.

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