Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Why verb-initial languages are not frequent

by

Andre Grüning

Preprint no.: 10 2003

Why verb-initial languages are not frequent

André Grüning*
Max-Planck-Institute for Mathematics in the Sciences
Inselstr. 22–26
D-04103 Leipzig, Germany

February 7, 2003

Abstract

In our simulations with simple recurrent networks we demonstrate that small artificial languages are learnt worse or better depending on their basic word order. We show that verb-initial languages are difficult to learn, reflecting the lower frequency of verb-initial natural languages.

We try to go beyond mere simulations proposing two objective mathematical measures to explain our results.

1 Introduction

Basic Word Order Most natural languages can be assigned a basic word order, the order in which verb `V`, (non-pronominal) subject `S` and a possible (non-pronominal) direct object `O` appear in simple declarative sentences. English e.g. is an `SVO`-language, while Welsh (generic: "Lladdodd y ddraig y dyn." / *killed the dragon the man*, i.e. "The dragon killed the man.") is `VSO` and Japanese is `SOV` (generic: "Gakusei-ga hon-o yonda." / *student book read*, i.e. "The student read a book.")).

The six possible orders of `S, V, O` are not equally frequent in the world's languages [13], see table 1.

Connectionism We want to follow the connectionist approach assuming that complex (linguistic) behavior can be explained better by sub-symbolic computation using neural networks rather than symbolic rules.

While traditional linguists ascribe the similarity of natural languages to some innate hard-wired *universal grammar* (UG), the connectionist belief is that rule-like behavior emerges from the cooperation of many simple neurons.

*e-mail: `gruening@mis.mpg.de`

order	frequency	$\mathbf{H(3|2) - H(3|21)}$	states
SVO	42%	0	10
SOV	45%	0.218	10
VSO	9%	0.817	15
VOS	3%	0.820	15
OVS	$< 1\%$	0	8
OSV	$< 1\%$	0.193	10

Table 1: Information loss as difference of conditional entropies and the number of states of the minimal FSA.

There has been a lot of work to show that UG rules need not be hard-wired but emerge in a natural way in trained neural networks. These network simulations are paralleled to natural linguistic behavior: Networks learn a particular rule better (worse) which is a hint why this rule is (not) preferred for natural language, too. This is what we do for basic word order in section 2, for subjacency see [4].

In fact, one only shows that a rule is learnt better/worse by this one particular network type with this one particular learning rule. Sometimes one feels the need to have a deeper explanation for this, making the connection to natural language stronger. This does not mean that we want to go back to formal grammar. We rather think in terms of dynamical systems, compare [11, 1].

Processing language means translating hierarchical structured data to a time series and vice-versa. It is our conviction that natural measures of complexity for times series can be found that are relevant for natural language. In the best case we hope these measures assign a low complexity to rules that are frequent in natural languages showing that not innate principles but more general natural principles form natural languages.

Some steps in this direction are undertaken in section 3.

2 Simulations

Simple recurrent networks (SRN) [5] are a simple type of recurrent artificial neural networks. They have an explicit short-term memory ranging back one time step, but develop during training a short-term memory that can implicitly extend further back in time [6].

Lexicon Our lexicon consists of a small number of verbs and nouns, see tables 2 and 3. For the sake of simplicity we do not model inflection or articles. Each entry can have special properties; *cry* e.g. takes no object and as subjects only those nouns that denote human beings, whereas *break* does not impose any restrictions

either on its subject or object. The labels in the lexicon are chosen arbitrarily, but in a way resembling their real world counterparts.

Grammar In contemporary grammar theory, the verb as the predicate is considered the most important part of a sentence [3, 12] and assumed to select its arguments (here subject and object). Even though a certain subject may restrict the possible objects of a verb and vice-versa, as a first approximation the verb is regarded to select its arguments independently.

Building a sentence To build a sentence a verb is chosen (with a chance of 1 : 10 as there are ten verbs), then its arguments are selected with appropriate chances according to the subcategorizing properties. The probability for optionally transitive verbs to take an object is 0.5. 182 different sentences can be generated, each with a certain probability. Two examples are given in table 4. We should rather speak of sentence templates than sentences as the word order is not yet defined.

Building a corpus A corpus of templates is constructed choosing 10000 sentence templates according to their probability and output in the six possible basic word orders to give six differently ordered corpora (`SVO, SOV, ...`). To each sentence an end-of-sentence marker is added.

Training SRN The network consists of input and output layer and one hidden layer. There are input and output neurons corresponding to each word and the end-of-sentence marker. In our case there are 17 each.

The corpus is presented to the network word by word using unary coding, and its task is to predict the next word by activating the corresponding output neuron. As there is an ambiguity in what the next word will be – there are sentences starting with the same words but ending in different ones –, what the network really will learn to predict is the probability distribution of possible next words in the context of the preceding ones. As usual, to check if the network

label	**property**
book	-
dog	a
house	-
man	h
mouse	a
woman	h

Table 2: Nouns in the lexicon. Properties: a = animal, h = human, - = none

label	transitive	subject properties	object properties
break	optional	-	-
call	optional	h	a $\vee$ h
chase	yes	a $\vee$ h	a $\vee$ h
cry	no	h	-
destroy	yes	-	-
eat	yes	h	a
kill	yes	a $\vee$ h	a $\vee$ h
move	optional	a $\vee$ h	-
run	no	a $\vee$ h	-
see	yes	a $\vee$ h	-

Table 3: Verbs in the lexicon. Required argument properties: a = animal, h = human, - = none

has succeeded in learning, the mean square error (MSE) between the networks' output activations and the probability distribution in the corpus is computed and used as the error signal for Elman backpropagation [5].

For each of the six corpora 100 Urns are initialized and trained for 100 epochs with a learning rate of 0.2. The nets turned out not to be very sensitive to variation in learning rate or a momentum different from 0. The whole experiment is repeated for nets with sizes of the hidden layer between 5 and 100.

Results The averaged MSE for 100 networks with 9 hidden neurons are printed in figure 1 up to epoch 20 (there are no qualitative changes after 20 epochs). This curve is generic for networks with a size of the hidden layer between 5 and 20. For the sake of clarity confidence intervals have been left out.[1] For networks with less than 5 hidden neurons, the errors become exceedingly high, so they fail to learn the languages at all.

We observe that the verb-initial languages are learnt much worse than the subject- or object-initial ones, furthermore `VOS` is learnt better than `VSO`.

The subject- and object-initial languages are learnt almost equally well: `OVS` performs slightly better than `OSV` and `SOV`, and `SVO` slightly worse.

[1]The statements in this section are with confidence of 95% or better.

sentence	probability
man eat mouse	1/40
house break	1/120

Table 4: Two sentences from the corpus in `SVO` order and their probabilities.

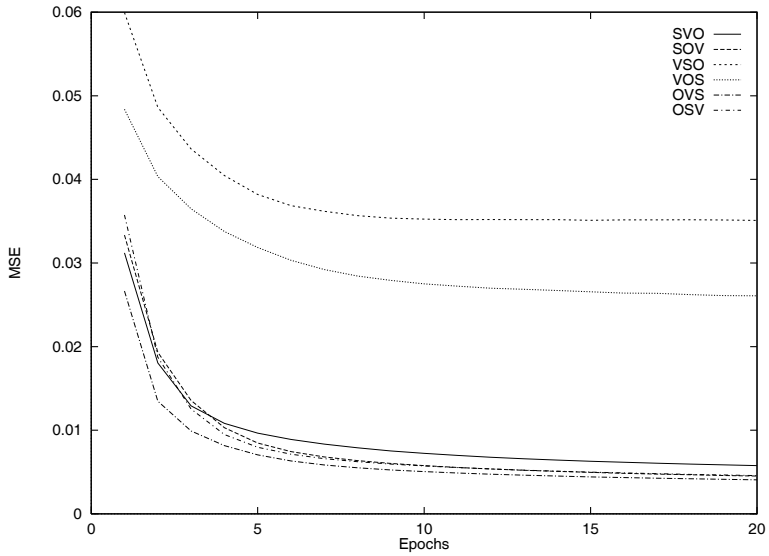

Figure 1: Simulation results, Elman network with 9 neurons in the hidden layer

Examining the output activations more closely, we note that the networks fail to learn to look back in time for more than one time step *accurately enough*: In a verb-initial language e.g., after the first noun the end-of-sentence marker is always activated to a small but above-background extent, irrespective of whether the verb is transitive or not. Similarly in SVO, the networks fail to distinguish subject and object position predicting to some degree another verb to follow after the object or end-of-sentence already after the subject.

For networks with more than 20 hidden neurons the differences between all the languages diminish more and more, whereas the MSE do not fall significantly but remain constant at about 0.005. Since with the size of the hidden layer the short-term memory capacity of the networks grows, why nets with a big hidden layer neurons have less difficulty to learn the verb-initial languages.

3 Measures of Complexity

Entropy An objective measure for the amount of information contained in a set of symbols (e.g. letters, words, sentences) each appearing with certain probability is the information entropy [8].[2] The higher the entropy the more difficult it gets to predict the next symbol correctly.

Using the probabilities of the 182 different sentences in our corpus the set of

[2]We calculate the entropies using log_2, then the entropy equals the average number of bits for identifying a single symbol, if an optimal code were used.

sentences has an information entropy $H_0 = 6.814$. It holds

$$H_0 = H(1) + H(2|1) + H(3|21),$$

where $H(1)$ is the entropy for predicting the first word, $H(2|1)$ the conditional entropy for predicting the second word knowing the first one, and finally $H(3|12)$ denotes the conditional entropy for predicting the third word knowing both the first and the second one.

Assuming that it is harder to keep more items in mind a longer time (this applies to SRNs and to human beings), an interesting question is how much information we lose, if we can look back in time only one step. For each language type we have therefore calculated $H(3|21)$ and the entropy $H(3|2)$ for the third word, knowing only the second one. The difference $H(3|2) - H(3|21)$, i.e. the loss of information is printed in table 1 (see [9] for similar ideas).

`SVO` and `OVS` are in this sense optimal as no information is lost reflecting the fact that `S` and `O` are selected independently. Knowing `V` and additionally `S` does not give more information about `O` than knowing `V` alone, and vice-versa.

The verb-initial languages have the biggest information loss, making them more difficult to predict. The information loss for `SOV` and `OSV` is smaller.

Counting States To the degree considered here, basic word order is a linear phenomenon, i.e. no recurrence is involved. Our languages can be produced and recognized by finite state automata (FSA) [10]. It has been shown that neuronal networks simulating an FSA must develop a representation of the states of the FSA [2] in their state space[3]. This implies that regular languages with a smaller minimal FSA are learnt easier as the lower number of states imposes fewer constraints on the state space dynamics of the neuronal network. [4]

Whereas the minimal automate for subject and object initial language have a comparatively low number of states, verb-initial language require more states (table 1). This again is due to the fact that the verb selects its arguments. So if the verb comes first more information (i.e in `VOS` about the subject) has to be stored a longer time (as the subject intervenes verb and subject), see figures 2 and 4.

4 Conclusion

We derived from a small lexicon a simple corpus of two and three word sentences. This corpus with six possible different word orders was fed into Elman networks as a word prediction task. We computed conditional entropies and the minimal FSA for each language.

[3] But it is not guaranteed that the minimal FSA is learnt, see e.g. [7]

[4] We belief other factors to be involved, too, as e.g. the number and similarity of transition between the states, the number of different input symbols and so on.

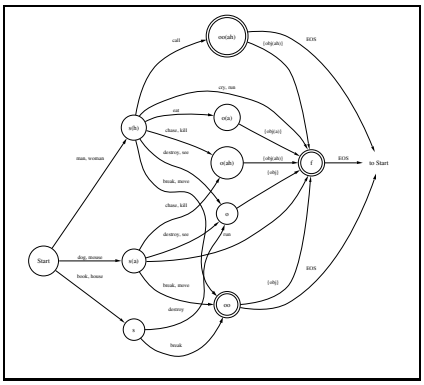

Figure 2: Minimal Automaton for SVO, and the VOS language respectively.

The computer simulations, entropies and the FSA demonstrate that verb-initial languages are more difficult to learn or have a higher complexity than argument-inital ones. The precise order within this groups varied, but is more similar for the members within each group than to any language outside the respective group.

Leaving aside the object-first languages (OSV, OVS, VOS) for a moment, the simulations as well as the additional considerations about entropy and FSA reflect the frequency distribution of word orders in the world's languages. Our theoretical consideration back-up the simulations and yield a measure of complexity that is indepent of the particular network type and learning rule.

Why did our approach fail for the object-first languages? It is clear from the construction of our corpus, that object and subject are treated almost symmetrically. Thus results should be comparable when subject and object are exchanged.

But why are object first language so rare then in the real world? Our setup is such that only syntactic phenomena can be captured. Invoking now pragmatics we argue here that in real languages it might be useful to include subjects, which often give the topic, in the beginning of a sentence to enable the early use of

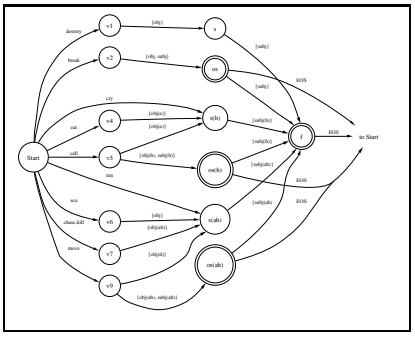

Figure 3: Minimal Automaton for VOS

contextual information, breaking the symmetry between subject and object.

References

[1] Mikael Bodén, Janet Wiles, Bradley Tonkes, and Alan Blair. Learning to predict a context-free language: Analysis of dynamics in recurrent hidden units. In D. Willshaw and A. Murray, editors, *Proceedings of the Nineth International Conference on Artificial Neural Networks (ICANN 99)*, pages 359–364, 1999.

[2] Mike Casey. The dynamics of discrete-time computation, with application to recurrent neural networks and finite state machine extraction. *Neural Computation*, 8:1135–1178, 1996.

[3] Vivan J. Cook and Mark Newson. *Chomsky's Universal Grammar*. Blackwell, Oxford, 2^{nd} edition, 1996.

[4] Michelle R. Ellefson and Morton H. Christiansen. Subjacency constraints without universal grammar: Evidence from artificial language learning and connectionist modeling. In *The Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, 2000.

[5] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[6] Jeffrey L. Elman. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7:195–225, 1991.

[7] C. L. Giles, C. B. Miller, D. Chen, H. H. Chen, G. Z. Sun, and Y .C. Lee. Learning and extracting finite state automata with second-order-recurrent neural networks. *Neural Computation*, 4:393–405, 1992.

[8] Stanford Goldman. *Information Theory*. Prentice Hall, New York, 1953.

[9] Peter Grassberger. Toward a quantitative theory of self-generated complexity. *International Journal of Theoretical Physics*, 25(9):907–938, 1986.

[10] John E. Hopcroft and Jerrey D. Ullmann. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Mass., 1979.

[11] Cristopher Moore. Dynamical recognizers: Real-time language recognition by analog computers. *Theoretical Computer Science*, 201, 1998.

[12] Carl Pollard and Ivan A. Sag. *Head-driven phrase structure grammar*. Univ. of Chicago Pr., 1994.

[13] R.S. Tomlin. *Basic Word Order: Functional Principles*. Croom Helm, London, 1986.