OPEN ACCESS

Vocabulary Learning and Instruction
ISSN 2981-9954
Volume 13, Number 2 (2024)
https://doi.org/10.29140/vli.v13n2.1502

Castledown

Vocabulary Networks Workshop 4: Changing the Connections in a Vocabulary Network

Paul Meara
Swansea University and University of Oxford

Imma Miralpeix
University of Barcelona

Abstract

This paper is part 4 in a series of workshops that examine the properties of some simple models of vocabulary networks. This Workshop explores how the overall activity level of a vocabulary network can be altered by changing the connections in the network (i.e., by implementing relinking events). The Workshop is linked to an online practice room where readers can explore these processes for themselves.

Introduction

So far in these workshops we have been discussing some simple models of how a vocabulary might work. These models are basically a network of words; each word in the network has two possible activity states (ON or OFF); each word has links to two other words in the network; and each word responds to the inputs it receives from these linked words in a specified way. In this workshop we are going to look more carefully at the links between words, but before we start this exploration, we need to give a bit more consideration to the nature of these links. In the previous workshops we have talked about "the input that WordA receives from WordX and WordY" – the two words that WordA is linked with. This wording is a bit biased in that it implies that WordX and WordY actively provide a specific input to WordA, however, and it leads us to ask questions like: why does WordX choose to send an input

Data Availability Statement: All relevant data are within this paper.

to WordA, and what does this input signal consist of? The problem here is that we are focussing on the relationship that WordX and WordY have with WordA, rather than on the relationship WordA has with other words in the vocabulary. On reflection, a better way of describing this relationship is to say that WordA is linked to WordX and WordY, that WordA monitors the activity state of these two words, and changes its own activity state accordingly. Conceptually, this is not a big change of emphasis, but it does slightly change the way we talk about the links between words in the network. The new wording makes it clear that WordX and WordY do not really "care" about WordA. They broadcast their current status at large, but only a word that monitors the status of WordX and WordY will respond to this information. This will become important in later workshops when we talk about the way networks develop and grow.

It is worth pointing out here that *monitor* has two main meanings in English. In one sense *monitor* implies some kind of supervision and direction. Krashen (1982) uses the word in this sense in his Monitor Model of L2 acquisition. *Monitor* also has another sense equivalent to "observe". In this workshop, we will be using the word in this sense, so we will say, for example, "WordA monitors the activity state of WordX and WordY", or "WordX and WordY are monitored by WordA". And words which are being monitored will be referred to as "watched words", as in "WordA has two watched words, WordX and WordY."

In Workshop 2 and Workshop 3 of this series, we looked at two ways of increasing the overall level of activation in a vocabulary network. In Workshop 2 we found that just turning words ON sometimes led to a temporary uplift in the overall level of activation, but it was necessary to repeatedly turn many words ON to achieve this uplift, and the effects seem to dissipate once this forced activation stops. This suggests that merely turning words ON is not in itself an effective way of increasing the overall level of activation in a vocabulary. In contrast, in Workshop 3, we saw that changing the way just a few words respond to the words they monitor often results in very large uplifts in the overall level of activation in a network, and this suggests that lexical fluency might be a key feature of lexical competence. We also noted that this particular feature has only rarely been studied as part of the vocabulary research enterprise. Furthermore, we found that networks where about half of the words were easy to turn ON were especially sensitive to small changes. The main conclusion we drew from these simulations is that activity in a vocabulary network is a property of the network as a whole, not just a feature of the individual words that make the network up. In order to raise the level of activity in a network, you have to change the underlying structure of the network, not just the superficial characteristics of its component words. This is an important conclusion because it suggests that teaching activities that focus on a few individual words might be missing the point.

In this fourth Workshop, we will examine how the overall level of activity in a vocabulary network can be altered by relinking events – events which change the connections in a network. You will recall that in the models we are using in these workshops, each word in our model vocabulary networks monitors just two other words in the network, and whether a word is ON or OFF at any particular time depends on two word-level features: the activity state of the two watched words and how each target word responds to these inputs. In Workshop 3 of this series, we experimented with changing the Response Type of words (i.e., making words easier to activate by

changing AND words into OR words). In Workshop 4 we will look at the effect of changing the links that connect the words in a network together.

Another Approach to Activating a Network

In all the models that we have looked at so far, each word in the vocabulary network has been randomly connected to two other words. However, we have also seen that different connections result in different levels of activation. In the simulations we have worked with so far, you were able to choose the value of the NTWK parameter and, whenever you do this, you get a network which is connected together in a unique way. All other things being equal, different values of the NTWK parameter result in networks with different attractor states. This suggests that changing the connections in a network might cause its overall activation level to change, though it is not immediately obvious what the direction of these changes will be. We will explore the effects of changing the connections in a vocabulary network in this episode of the Workshop.

As usual, we need to think carefully about what we mean by "changing the connections in a vocabulary network" before we rush into a new set of simulations. One easy way to examine the effects of changing links is simply to select one target word, and swap both of its current links for two new ones each time an event is called. To do this we need to do something like the algorithm shown in Figure 1. This algorithm appears to be quite straightforward, but you should notice that it makes a number of assumptions that might or might not be critical to the way our simulations work.

The first assumption is that an event changes both of the words that are monitored by WordA. It would be more conservative for an event to change just one link, but for the moment it is easier for us to work with a more radical relinking where both links are replaced.

The second assumption is that any word can be relinked by a relinking event: the program chooses a target word and two new links at random, and it does not care what the current activation state or the response type of these words might be. However, the program does turn ON any word that is involved in an event: the target word and both of the words it now monitors are all turned ON when an event takes place.

This approach has some important consequences. Every event will usually activate three words in a network where the overall level of activation is low, but it may activate fewer words in a network where the level of activation is already high because most words will already be ON. This seems like a reasonable set of minimal assumptions to make in the first instance, though it is worth thinking about how the simulations would work if we placed some constraints on which words might be eligible for relinking, or if we imposed some constraints on the choice of words which can serve as words for our selected target to watch. We will examine some of these variants later, but for the moment, as usual, we will keep things simple and avoid making things overly complicated.

The third assumption that the algorithm makes is that any word in a vocabulary network can be selected as a word for our target word to monitor, and for the moment at least, we will be choosing these new watched words at random. Again, the rationale

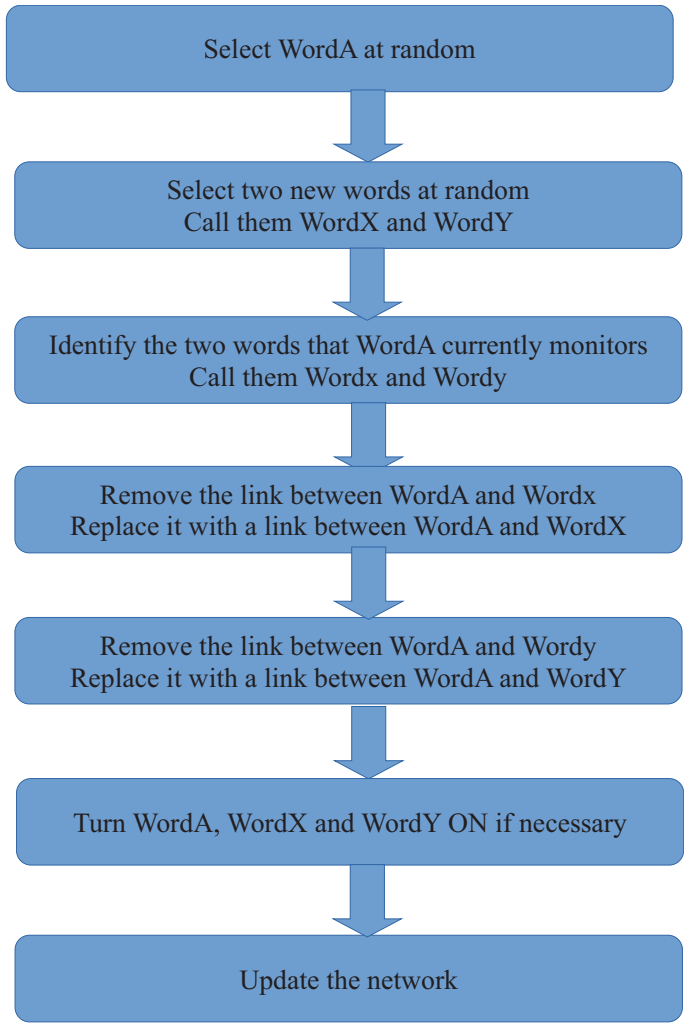

Figure 1. *The structure of a relinking event. WordA is the target word for this event. Wordx and Wordy are the two words that WordA monitors before the event. WordX and WordY are the two new words that WordA monitors after the event.*

for this decision is that random connections are simpler to implement than more constrained connections. It is obvious, of course, that a system of random connections between words is not likely to be a good model for how a real vocabulary works. Leaving aside the obvious criticism that a real vocabulary is likely to use semantic or phonological connections rather than random connections, there are other reasons why we might expect random relinking events to generate unpredictable changes in a vocabulary network. A couple of examples will make clear why this might be a problem.

Consider first a network where WordA currently monitors two words,Wordx and Wordy. If WordA is an AND word that requires both of its watched to be ON for it to be activated, and if WordA is currently active (ON), then replacing both of its links MIGHT change its activation status, but any relinking event will not immediately affect WordA's activity status. Suppose, for example, that both of WordA's watched words are already activated, so that WordA is also currently ON. Running the algorithm in Figure 1 will always result in WordA remaining activated in the short term, since step 6 of the algorithm turns all three words involved in the event ON, regardless of their current status. However, the real question is whether this activity will turn out to be long-lasting. This will depend on the current status of both the new links WordX and WordY. Both of these words will be activated by step 6 of the algorithm. However, if either of these words is normally OFF, and reverts to that state when the network is updated, then WordA will quickly return to the OFF state too. The chances of this happening will depend on the overall activity level in the network: if the network contains only a few activated words, then the chances of WordX and WordY remaining activated will be small, and this means that WordA is also likely to become de-activated. On the other hand, if the network contains a lot of activated words, then the chance that the program will randomly select two already activated words will be higher, and this means that WordA will be likely to remain in an activated state. When the number of active words in a network is about 50%, then a random relinking event will sometimes turn WordA OFF, and sometimes will leave its status unchanged. However, the probability that WordA will NOT be permanently activated by an event is higher than the probability that WordA WILL be activated by an event (as there are four possible patterns of activity if we select inputs at random, and only one of them results in both inputs remaining ON after the event). If Word A is OFF when the event takes place, then these arguments work in reverse – WordA and the two new watched words, WordX and WordY, will all be activated by the event, but whether WordA remains activated will depend on whether the program selects two inputs which are already activated or not. Unless the number of ON words in the network is very high, target words are more likely to be turned OFF by a random event that swaps two links for two new ones.

Similar arguments apply if WordA is an OR word, which needs only one of its watched words to be ON for it to be activated. In this condition, a random relinking event is more likely to be effective, resulting in the activation of WordA, but even here, the chances of finding a single random connection that can activate WordA will depend on the overall number of activated words in the network. The chance that a relinking event will select at least one activated word for WordA to monitor will again depend on the number of activated words in the network. Again, with each word in the vocabulary monitoring two watched words, there are four possible input patterns, but three of these will result in the activation of WordA. So, over time, OR words should be more likely to become activated than AND words, particularly when the number of activated words in the network is already high. What are the implications of this bias?

Either way, it looks as though implementing the algorithm in Figure 1 will probably result in an overall increase in the level of activity in the network. However, this will only work in the long term, and progress will be affected both by the number

of OR words in the network and by the number of activated words that the network contains. Overall, we can expect that a network which experiences random relinking events is going to be relatively unstable, and it ought to be very difficult to systematically and permanently increase the number of activated words in a network using events of this type. Despite these reservations, it is still worth our while to investigate random relinking events, so that we can develop a feel for the way random changes affect the performance of a vocabulary network before we start to look at re-linking events that are not random. We will do this using Program-7.

Program-7: Random Relinking

You can run Program 7 by going to the Workshop Home Page and clicking on the Program-7 button: **https://www.lognostics.co.uk/Workshop/index.htm.**

The data input screen for Program-7 contains only three parameters that you can vary. All three parameters will be familiar from earlier programs in the Workshop. **NTWK** determines the basic structure of the network, how its words are linked together and how each word responds to inputs from the two words it monitors. The **nEv** parameter determines the number of events that your network will experience. Events can occur between update 100 and update 900. The **rEv** parameter[1] determines which words are affected by an event – which words will have new watched words to monitor, and what these new watched words will be.

Figure 2 shows the output of Program-7 with parameters NTWK: 1250, nEv: 20 and rEv: 1235. As usual, the green line in the chart indicates the number of activated words in the network after each update, and the red dots at the bottom of the chart indicate that an update event has taken place. This combination of parameters should

[1] **How the rEv parameter works**

In this workshop, each event selects a target word whose watched words need to be changed. Each event identifies three words: the first word is the target word for this event, and the other two words are the new words that the target word monitors.

When the programs initialise, they generate a large set of random numbers. The make-up of this set is determined by the rEv parameter. Every value of rEv gives a different sequence of random numbers between 1 and n, where n is the number of words in the model vocabulary. So, for example, rEV: 2711 might generate a sequence that looks like this:

642, 178, 359, 805, 221, 510, 998, 72, 431, 693, 114, 887, 546, 920, 37, 260, 409, 102, 835, 956…

When an event is called, the program chooses the first three words in the sequence. With this sequence, it finds Word642 and tells Word642 to monitor Word178 and Word359 instead of the words that it is currently monitoring. These three numbers are then discarded. The next event will find Word805, and tell it to monitor Word221 and Word510.

A different value for the rEv parameter (1946, for example) might generate a sequence that looks like this:

13, 482, 79, 291, 617, 57, 324, 908, 153, 862, 45, 201, 739, 64, 987, 305, 190, 528, 416, 823…

In this case, the first event will affect Word13, the second event will affect word 291, and so on.

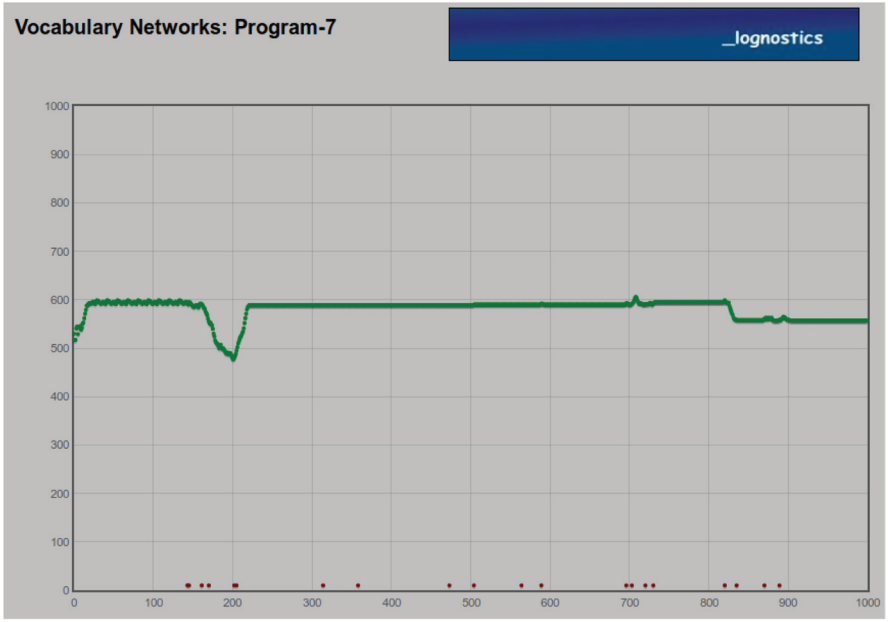

Figure 2. *Random changes to the links between words. NTWK: 1250; nEv: 20; rEv: 1235.*

give you a slightly unstable network, where about 600 words are ON when the network reaches its attractor state.

There are a couple of points to note here. The first is that, overall, the twenty events have not made very much difference to the level of activity in the network. By update 50, this network is in a (slightly unstable) state where about 600 words are ON. After all the events have been implemented just a few words have been turned OFF. Nevertheless, it does look as though this change in activity is a permanent one. The second point to note is that most of the individual relinking events have no effect on the network at all – only a handful of the 20 events seem to generate a noticeable change in the activity level of the network, and in most cases these changes are temporary. Only one large permanent change is in evidence – at update 810. However, it is also worth recording that a small cluster of events occurring in quick succession does seem able to generate larger fluctuations in the network – see the events between update 150 and 200. Finally, we should note that the overall trend in this network is for a relinking event to turn words OFF rather than turning more words ON.

Next, you can raise the number of events in this simulation to 200 (reset the value of nEv to 200, but leave the other parameters untouched). The output from this simulation should look something like Figure 3.

Here, we see the effects of a long series of relinking events. Little change takes place as a result of the first events in the simulation (updates100–200) – though there is a small reduction in the overall activation level of the network. After that, there is a steady growth in the number of activated words in the network, with marked increases at

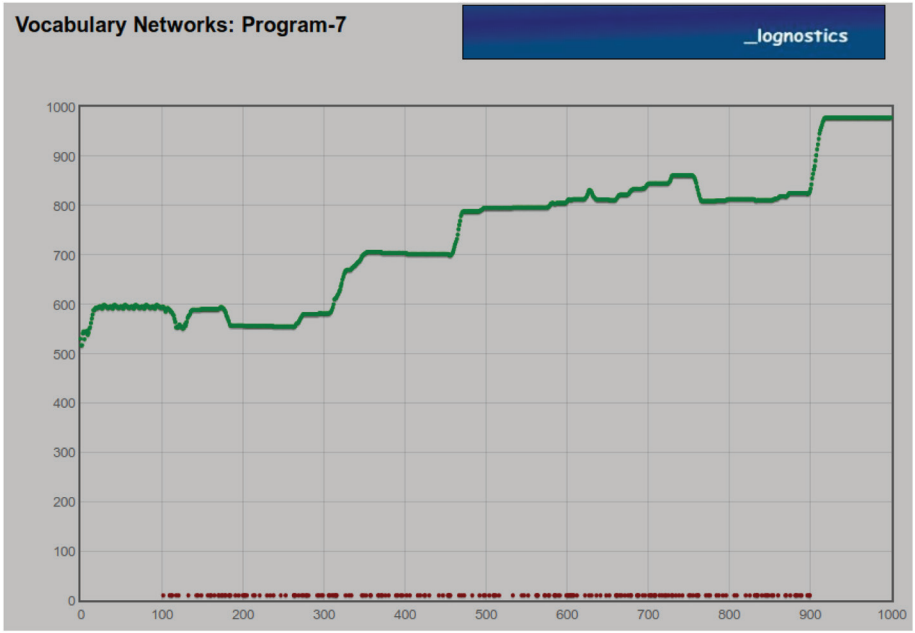

Figure 3. *Random changes to the links between words. NTWK: 1250; nEv: 200; rEv: 1235.*

update 310, update 450 and update 900. When the events stop, the network eventually settles into an attractor state where almost all of the words are ON. This might lead you to expect that networks always increase their overall activation level if they experience enough relinking events. However, if you experiment with different values of the rEv parameter, you will find that this is not always the case. Figure 4 shows an example of this.

Here the NTWK parameter and the nEv parameter remain unchanged, but the rEv parameter has been changed to 1239. The result is a model in which nearly all the events have a negative effect on the overall level of activation in the network. Once the relinking events stop at update 900, the network quickly settles into an attractor state where only 180 words are activated. This figure clearly shows that the rEv parameter plays a much more important role in Program-7 than it did in our earlier simulations.

You should experiment with other values of the rEv parameter and explore how much variation you get when different words are selected for relinking. The best way to explore these questions is to set the nEv parameter to a value that gives you a small number of evenly spaced relinking events (e.g., 10 to 20) and keep varying rEv. This will allow you to estimate the effects of single relinking events. You should find that events do not always have the same effect on the network. Some events will have minimal effects, while others will be larger. When you find a value of rEv that produces significant losses, test it out with different values of the NTWK parameter.

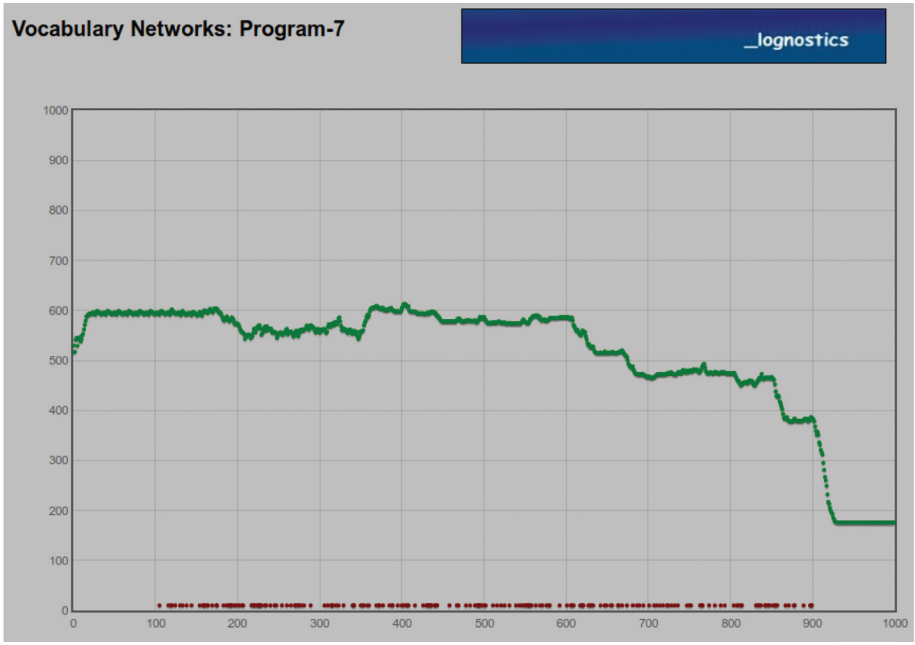

Figure 4. *Network with relinking events but with a low level of activation at the end. NTWK: 1250; nEv: 200; rEv: 1239.*

Do you still get significant losses? Ask yourself: how big are the changes that occur in a network when a relinking event takes place? Are these changes very large, or typically quite small? And does giving a word new watched words to monitor tend to make it more or less likely to be activated?

Next, you should ask how typical these findings are and see if you can make any generalizations (keep the evenly spaced relinking events, while changing the value of the rEv parameter). Ask yourself: how likely is it that a randomly selected set of relinking events will move your network into a new attractor state? How many of these relinking events increase the number of ON words by more than 20%? How easy is it to find a relinking that gives you an extreme result where almost all the words in the network are ON or almost all of them are OFF? How often do you get a very large shift where a predominantly active vocabulary moves into a state where nearly all its words are OFF?

You should also experiment with different values of the nEv parameter, and ask whether more events are more likely to result in an extreme outcome. For example, you might want to run a set of models with 50 events, 100 events, 150 events and 200 events and ask: does the probability of a large shift in activation increase as the number of events increases? Is there a threshold beyond which you always get a significant shift in activation? Are these shifts positive or negative? What does this tell you about the potential of relinking events to cause a permanent increase in activation levels?

What you should find is that random relinking events are very unpredictable in the short term. In the longer term – when you set nEv to give you more than 100 events, for example – there are some general trends in the simulations. Networks where the overall level of activation is high will generally tend towards higher levels of activation, and networks where the overall level of activation is low will generally tend to drift towards an even lower level of activation. Networks that have close to zero activation will rarely move to a higher level, but networks that are almost fully activated can sometimes lose a lot of activation. Networks where 40 to 60 percent of the words are activated seem to be inherently unpredictable: they tend to remain roughly at the same level of activation, but experience both ups and downs. Most of these results fit with how we would predict a network to behave. Probably the most important feature of a network in this simulation set is the number of active words in the network at the time an event takes place. However, the tendency to move towards an extreme value can be overridden by our random choices, with the trend suddenly changing direction for no apparent reason. Once again, this feature seems to be especially sensitive to the value you give to the rEv parameter.

You should also find that the relinking events in this Workshop are much less predictable in their effects than were the events in Workshop 2 and Workshop 3. The obvious interpretation of this is that not all words are equal when it comes to replacing their current watched words: some words can be relinked with impunity, but relinking other words can sometimes have a massive effect on the overall activity level of the network. It is not immediately obvious what causes these disparities. Again, these results might lead you to ask a number of questions about the detailed structure of the network, and how small changes to the parts of the network around one or two words can resonate on a larger scale. This might lead you to ask: what is special about the words that trigger large changes? Do real vocabularies have similar characteristics?

We have provided some further questions to guide your explorations of relinking events. As usual, there are no correct answers to these questions. They are designed to make you think more critically about what these simulations are doing, and to help you better understand the other simulation sets in this Workshop.

Some Questions to Ask:

- *How many different types of outcome can you identify in this simulation set?*
- *Can you predict what the outcome of a simulation will be from its initial attractor state?*
- *How often does a network return from the dead after most of its words have been turned OFF?*
- *What are the chances that a highly activated network will stay that way?*
- *How big is the window that generates unstable activation levels?*
- *Are the activation patterns affected by the frequency of the relinking events?*
- *How big is a typical relinking event? (You can investigate this by setting the nEv parameter to a very low value, e.g. nEv: 5 or nEv: 10, so that relinking events are fairly rare. Do these rare events make much difference to the overall activation levels in the network?*

- *What proportion of these events have a noticeable effect on the activation levels?*
- *How often does a really large effect occur? Are there any special conditions that make these large effects more likely?*
- *In this simulation set, a relinking event replaced* ***both*** *of the words monitored by a selected target word, but we could have programmed events where only one of the existing watched words is replaced. What effect would this have had on the performance of the models?*
- *What might trigger a relinking event in "the real world"? How often would you expect such events to occur?*

Program-8: Non-Random Relinking

It is obvious that Program-7 was not a realistic way of modelling the development of L2 vocabulary – random relinking just does not look like a plausible mechanism for vocabulary growth, and the results of your own explorations should largely have reinforced this belief. Nonetheless, this simulation set throws up a number of interesting and perhaps unexpected ideas about the way a vocabulary network might behave when it experiences relinking events. First, we have found once again that an apparently simple event may have different consequences depending on the overall state of the network on which it operates. Critically here, a random relinking event has different effects depending on whether the words in the vocabulary network are predominantly ON or predominantly OFF. If a network is in one of these states, then a random relinking event will tend to reinforce this characteristic. However, if the network does not have a clear majority of words ON or OFF, then the random relinking events can push it in either direction, or the network may oscillate up and down around its initial resting state. Second, you should also have found that networks can sometimes be subject to very large swings in their activation level – most relinking events have quite small results, but occasionally we get a relinking event which has a very large effect on the overall activation of the network. Clearly, not all relinking events are "the same" in this respect.

The main lessons that we need to draw from these observations is that vocabulary networks are extremely sensitive to the way words are linked together, and that left to themselves a sequence of relinking events can have unpredictable (and possibly unwanted) consequences. The obvious conclusion is that the relinking events we have explored in the previous sections are unlikely to provide a reliable way of raising the level of activity in a model vocabulary network.

However, before we accept this conclusion at face value, it is worthwhile looking at other ways in which we could implement a relinking event. Obviously, random relinking is a bit crude as a mechanism, but it might be worth looking at some more subtle relinking events to see whether they might have a positive impact on the structure of a network and the way it responds.

One simple way of improving on random restructuring events would be to design a simulation set where target words are relinked only to words that are already ON when the relinking event occurs (see Figure 5). This small change will almost certainly result in an increase in the number of activated words in a network, since we are deliberately selecting watched words which provide the activation necessary to turn

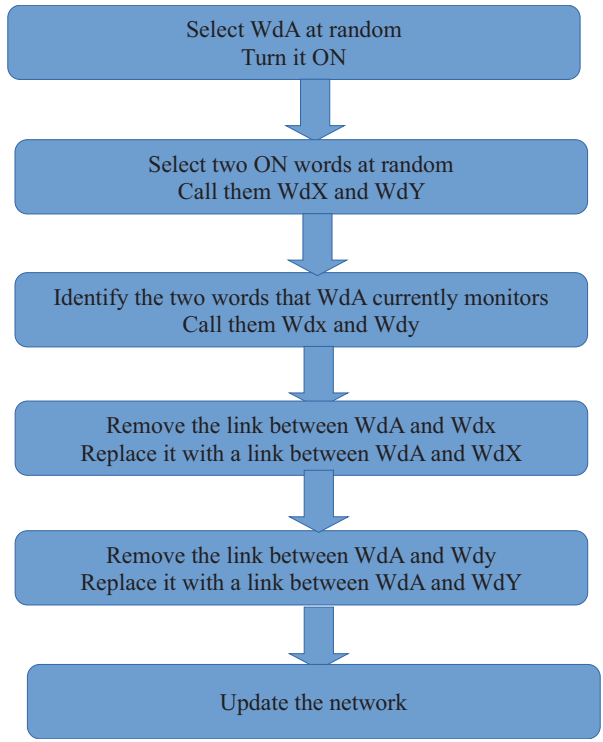

Figure 5. *The algorithm for Program-8.*

a target word ON. Indeed, if we insist that both of the monitored words are already activated, then even target words of the AND type (words which require both their monitored words to be ON for them to be activated) will be successfully activated. Normally this activation will be permanent: the only circumstance where activation might not be permanent would be if we selected input words which were themselves only temporarily activated (perhaps as a result of an earlier event), and could therefore revert to a non-activated state. This idea is explored in Program-8.

You can run the simulations by selecting Program-8 from the Workshop start page: **https://www.lognostics.co.uk/Workshop/index.htm.** As usual, you need to think about how these simulations will work out before you run them.

The data input page for Program-8 is identical to the input page for Program-7, except for a small change in the way the program is described. You have three variable parameters: **NTWK** controls the basic structure of the network; **rEv** controls which words are selected for updating, and which words are chosen as watched words for the target words; each event selects two new words for the selected target word. **nEv** determines how many relinking events take place in your simulation.

Figure 6 shows an extreme example of how relinking events might affect a network with a low level of activation. The network initially settles into a very low level of activation – fewer than 50 words are activated when the network reaches its attractor

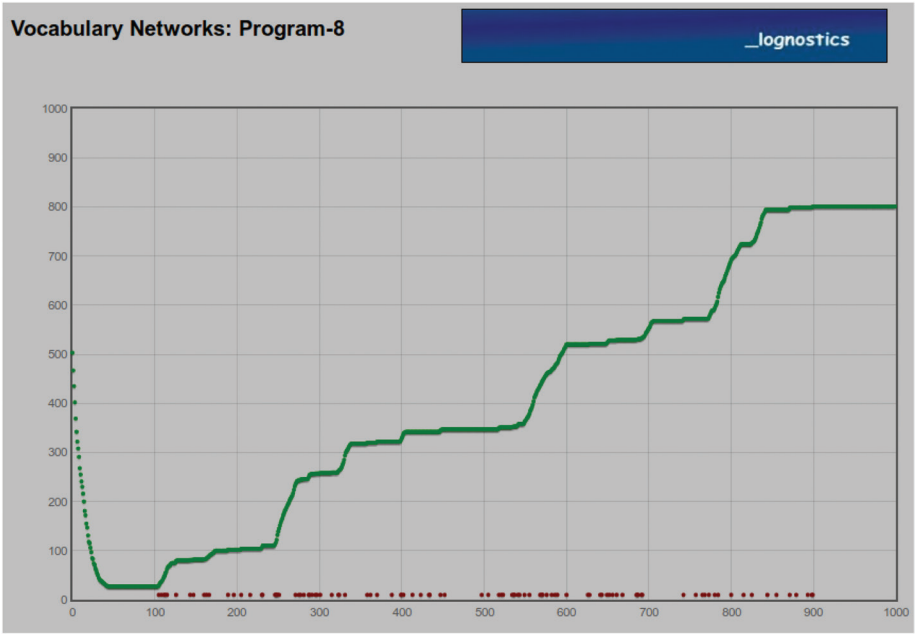

Figure 6. *Constrained changes to the links between words. NTWK: 1500; nEv: 100; rEv: 1234.*

state. It then undergoes 100 relinking events, almost all of which cause a permanent uplift in the overall level of activation in the network. When the events stop at update 900, more than 800 words are in the activated state. This means that although each relinking event formally affected only one word, on average, each event has activated about eight additional words in the network.

You can explore this effect by changing the value of the nEv parameter, and testing the effect of a smaller number of spaced events. For example, you could set the parameter values to NTWK: 1500, nEv: 10 and rEv: 1234. The outcome of this simulation is shown in Figure 7. Each relinking event causes an uplift in the overall activation of the network – by update 900, the number of ON words has grown to about 235 words. Most of these uplifts are small, but there is one relatively large uplift at update 575, where around 120 new words are turned ON by a single event. Overall, the 10 events lead to an uplift of around 185 words: the large uplift at update 575 accounts for 120 of these, while the nine smaller events account for the other 65.

At this point, you should be asking questions like these:

- *Why does relinking a single word result in the activation of several more words? What conditions would be necessary for this to occur?*
- *Why do most relinking events result in small changes, while just a few events result in a lot more activation?*
- *How big can these large events be and how frequently do they occur?*

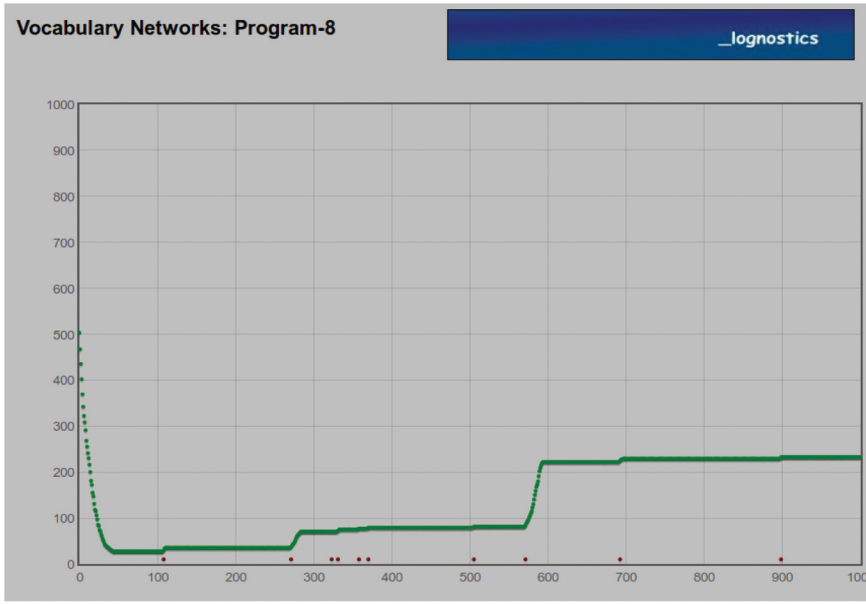

Figure 7. *Constrained changes to the links between words.*
NTWK: 1500; nEv: 10; rEv: 1234.

- *Are large activation events a particular feature of vocabularies with a low level of overall activation? Do they become less likely to happen as the number of activated words gets larger?*
- *How much variation is there in the patterns that you find?*
- *Are the patterns affected by the initial resting equilibrium of a network?*
- *How many relinking events are necessary for a network to reach a high level of activity?*
- *What happens when you get a series of relinking events taking place in quick succession?*
- *Typically, how long is the initial period of stability before the first jump in activity levels?*
- *How large is a typical jump in network activity?*
- *Occasionally, you will find that the overall level of activity in a network declines. What mechanisms might be causing this?*
- *How would the networks behave if only one of a target's watched words was being relinked by an event rather than both of them? Is this important?*
- *How would you expect the networks to behave if only one of the new newly selected watched words was required to be ON? Is this a reasonable constraint? How would you justify it?*
- *How do the outcomes of Program-8 differ from the outcomes generated by the programs we used in Workshop-3?*
- *In "real life" what might cause relinking to occur? How plausible is an event of this kind?*
- *What kind of exercises could you devise that might promote beneficial relinking for an L2 vocabulary?*

As usual, there are no right answers to these questions. Their main purpose is to help you think critically about the assumptions we have built into the simulations, and to identify other characteristics that you would like to model.

The outcome reported in Figure 7 seems to be fairly typical of the models in this simulation set. Relinking events which are constrained as they are here always increase the overall level of activation in a network, and, generally speaking, the resulting levels of activation are very high with all, or nearly all, of the words in the network moving permanently into an activated state. There are several points worth noting here. Firstly, the changes implemented by Program-8 almost always result in a permanent increase in the number of activated words in the network. Secondly, it does not take a large number of events to bring about a significant shift in the network's activation level. Thirdly, although each event directly affects only a single word, a number of other words may also be affected by the change. Fourthly, you might have noticed that the uplifts that occur with Program-8 are less dramatic than the ones we found in Workshop 3. Relinking target words to already ON words seems to provide a reliable but unspectacular way of raising the overall activation level in a vocabulary network. Finally, you should note that relinking events are less likely to induce a change in the overall activity of the network when the number of ON words is already large.

While the simulations using Program-7 suggested that relinking events were unlikely to be successful as a driving mechanism for change in a vocabulary network, the simulations using Program-8 strongly suggest that relinking words to already activated words can be a very powerful mechanism that can easily achieve high levels of overall activation. This is a good example of the way working with simulations can make us revise our theoretical views about vocabulary acquisition. Here we have a mechanism which looks to be incredibly powerful, but it is not one which has received much attention in the research literature. As usual, the main point to take away from this Task is the idea that we need to be very careful about the assumptions that we build into our simulations. Here, one small change to the way we have implemented the relinking events has clearly overturned the pessimistic conclusions we reached on the basis of Program-7.

Program-9: Other Ways of Modelling Non-Random Relinking

In Program-7 we saw what happens when words in a vocabulary network are relinked randomly, and we decided that random relinking was not an effective way of raising the overall activation level of the network. Random relinking was just as likely to lower the overall activation level as it was to raise it, and networks relinked in this way showed high levels of instability. In Program-8 we saw a particular type of non-random relinking, where words were relinked only to words that were already ON.

However, it is possible that a different kind of non-random relinking might be more effective, and in this simulation set we will look at a way of implementing relinking events that has some interesting implications for the behaviour of a network. You can explore this new method with Program-9.

In the earlier simulations in this Workshop, we assumed that all words are equally likely to serve as watched words for target words. When the networks are initialised, their watched words can come from anywhere in the network, so for example, in a

thousand-word network, Word50 might monitor Word49 and Word321, or Word99 and Word423. This way of initialising the networks leads to a rather flat structure. Although each word monitors two other words, this relationship is not symmetrical, and the randomisation of the network means that some words may be monitored by several other words, while other words will be monitored less heavily. The distribution of the monitoring links in the network is fairly "democratic", in the sense that most words will be monitored by one or two other words, but a few words will be monitored by more than two other words, and there will be a special category consisting of words that are not monitored by any other word. This last category looks like a new type of word which has no effect on the rest of the network. Structurally, this means that our networks will contain three types of words: (1) words that are not monitored by any other word – turning one of these words ON will have no effect on the overall activation level of the network; (2) words that are monitored by just one other word – turning one of these words ON will usually set up only a small ripple in the network; and (3) words that are monitored by a large number of other words – turning one of these words ON can potentially set off a large ripple of activation in the network. In Program-9, we ask what happens if relinking events impose a structure on the network by increasing the number of words in category (3) and at the same time increasing the number of words that do not serve as watched words. In Program-9, we implement this constraint by limiting the range of words which can be chosen as new links by a relinking event. Specifically, the program lets you decide how many words are available to be selected for monitoring.

You can run Program-9 in the usual way. Select Program-9 on the home page of the Workshop website: **https://www.lognostics.co.uk/Workshop/**. The input page is essentially the same as for the earlier programs, but in addition to the normal parameters **NTWK, nEv** and **rEv,** you have an additional parameter: **MaxN**. This parameter controls how tightly constrained is the selection range for new watched words in this simulation set. The lower this value, the more tightly constrained this range is. Setting the value of MaxN to 10, for example, means that all the new links implemented by an event will be chosen from the first ten words in the network. Setting the value of MaxN to 20 means that all the new links will be selected at random from the first twenty words in the network.

At first sight, this might seem like a trivial constraint, but it actually has some interesting knock-on effects. Basically, it means that words with low serial numbers will gradually increase their influence as more and more events take place. After some time, and many events, a large proportion of the words in the network will have connections to this central core of important words, and there will be an increase in the number of words that are not monitored by other words. This means that turning ON a randomly selected word in a constrained network is very likely not to have an effect on the network's overall activation level. However, if we activate a word that is one of the highly monitored words, then a lot of words will be affected by this change. Activation events – events that turn a single word ON – may generally not do very much, but occasionally a random event will activate a lot of other words that depend on the newly activated word for their own activation.

You can explore these ideas with the Program-9 simulation set. As usual, you need to think about how you would expect these new models to perform before you run them.

Figure 8 shows an extreme example of how a constrained network performs. Here we have a network that settles into an attractor state with a very low level of activation – about 25 words (update 50–100). Four hundred events are implemented. Each event relinks a randomly selected word to two new watched words chosen out of the first ten words in the network. The first few relinking events (updates 100–130) have only a minor effect on the overall level of activation in the vocabulary, but as more and more events are implemented the effects get steadily larger. There is a huge increase in the number of activated words around update 280, and shortly after this, the number of activated words is over 500. This peak is relatively short-lived, though, and with further relinking events, the activation level falls away again, decreasing to a much lower level around update 400. However, the network is now very unstable, and shortly after update 500 it experiences a massive uplift that has the effect establishing a new attractor state where ALL of the words in the network are ON.

If you have worked through the earlier programs in this Workshop series, you should find that a number of questions will immediately come to mind when you look at Figure 8:

- *Is the outcome of this simulation affected if we reduce the number of events?* You can answer this question by working with lower values of the nEv parameter. Start with a very low value for this parameter, 10 for example, and observe the effects of single events. You should find that events have a minimal effect on the network, and that most events do not involve a lasting change. Next, raise

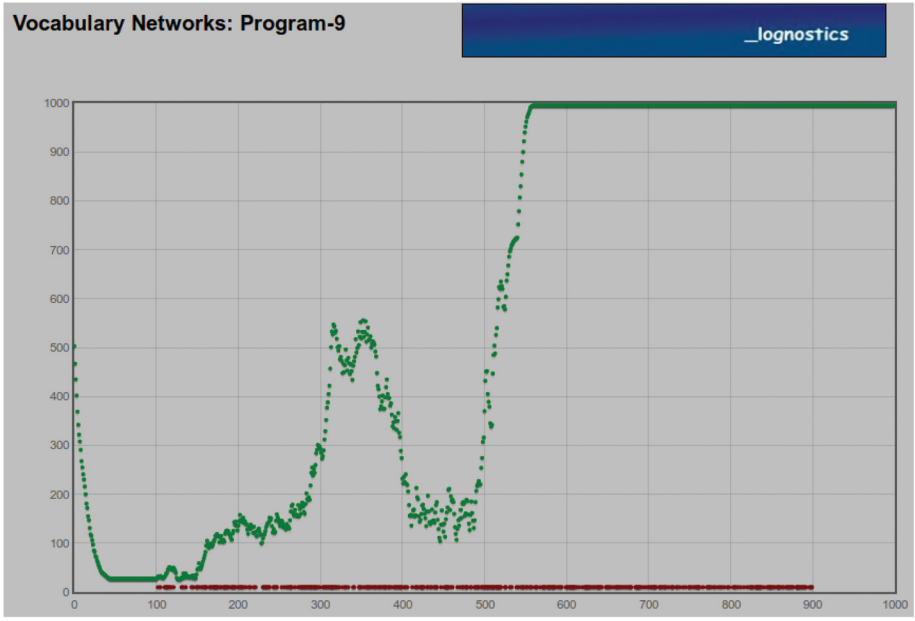

Figure 8. *A constrained network, where NTWK: 1500; nEv: 400; rEv: 1235; MaxN: 10.*

the value of nEv, and ask yourself what happens as events become more and more frequent. You should find that 305 relinking events are enough to trigger a huge uplift in activation.

- *Does this always occur, or does it depend on the particular words that are selected by an event?* You can examine this question by changing the value of the rEv parameter (e.g., try rEv: 1236 and rEv: 1234).
- *Do all networks eventually transition into an attractor state where all their words are activated?* Test this out by changing the value of the NTWK parameter.
- *Does it matter if we expand the range of words that can be selected as inputs by a relinking event?* Go back to the settings NTWK: 1500, nEv: 500, rEv: 1235 and experiment with different values of the MaxN parameter.

The answers to these questions will depend to some extent on your choice for the parameter values. Nevertheless, a number of generalisations do seem to be plausible. Firstly, it should be clear that the value of the MaxN parameter is very important in this simulation set. Smaller values of MaxN are more likely to result in a large activation uplift than larger values are: values over 100 rarely result in a massively activated network, though smaller values will sometimes lead to attractor states

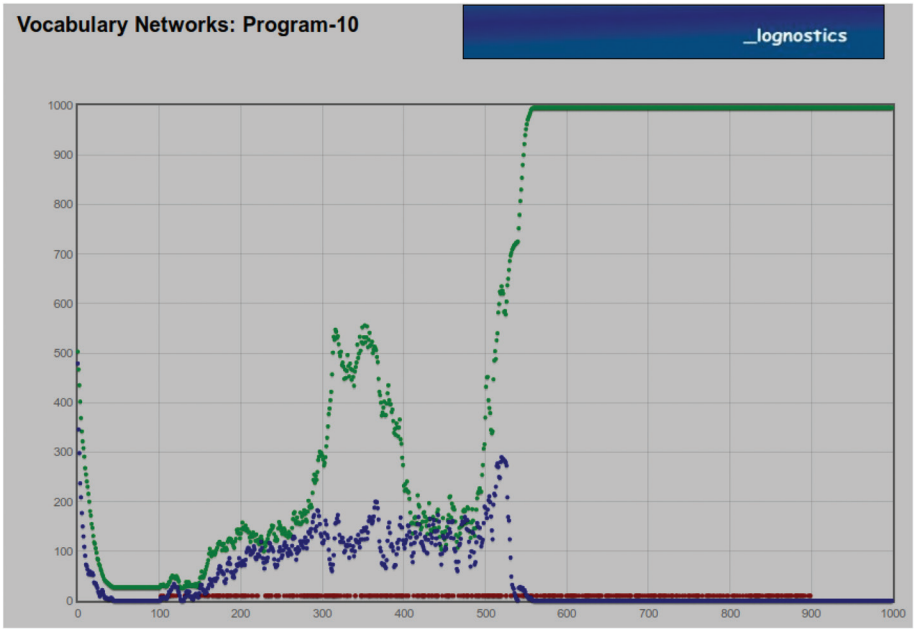

Figure 9. *A constrained network, where NTWK: 1500, nEv: 400, rEv: 1235, MaxN: 10.*
The green line shows the number of activated words at each update of the network.
The purple line shows that number of words changing their activity status from OFF to ON or ON to OFF at each network update.

where significant numbers of words are activated. Secondly, it should also be clear that the value of the rEv parameter strongly affects the outcome of these simulations. A combination of parameters that shows a strong effect will not always show a similar one with a different value of rEv, and this suggests, once again, that not all events are equal. Thirdly, it should be obvious that the shape of the graphs that we find in this simulation set look very different from the graphs that we reported in Workshop 2 and Workshop 3. With Program-9, the graphs tend to be discontinuous, rather than varying smoothly. This seems to indicate that a lot of words are changing between the ON state and the OFF state after each event.

Readers who are interested in this last idea can experiment with Program-10, which for space reasons we will not discuss in detail. All our earlier programs have simply reported the number of activated words in a vocabulary. Program-10 replicates Program-9, but it makes a more detailed report which tells you how many words are changing their activation status at each update. Figure 9 shows you an example of the output provided by Program-10, using the parameter values NTWK: 1500, nEv: 400, rEv: 1235 and MaxN: 10. Figure 9 replicates the output shown in Figure 8, but it has an additional data line that records words whose activity state is shifting from ON to OFF or from OFF to ON as a result of a relinking event. The point to note here is that the number of words affected by a single event can be extremely high. In Figure 9, for example, nearly a third of all the words in the network are changing their state at each update around update 500. If you want to explore this feature further, then a way to go is to use Program-10 to explore how much impact a single relinking event can have (work with low settings for the nEv parameter, so that you get a few spaced-out events).

Discussion

The main idea to come out of this simulation set is that the connections between words in a vocabulary network may be even more important than we have assumed so far. In a completely random network there are no constraints on how each word is connected to the rest of the network – apart from the constraint that each word monitors the activity state of two other words. With the programs in this Workshop, however, we have seen that some relatively small changes to this assumption can have very large knock-on effects on the behaviour of a vocabulary network. In particular, a network where we have a small core of words whose activity levels are monitored by the rest of the vocabulary seems to behave very differently from a network where monitoring is spread more evenly across the network.

There are some resonances here with Carter's idea of a "core vocabulary" (Carter, 1987). Carter suggested that some words in a real vocabulary might have a special status. The core vocabulary was defined as a small set of words with specific linguistic characteristics. Core vocabulary items a) substitute for a wide range of other words; b) tend to have antonyms; c) have a wide range of collocates; d) often have multiple meanings; e) tend to be super-ordinates; f) may be free of culture-specific uses; g) appear frequently in summaries of complex material; h) tend to be rated neutral in a semantic differential test; i) do not point to a specific field of discourse; and j) are not marked for tenor of discourse. However, it is not necessary for a core vocabulary

to have such a specific set of features, and other simpler, surrogate features might be used instead. For example, in a real vocabulary, most of Carter's features would be accounted for in terms of frequency of occurrence, or even age of acquisition (see, for instance, Nagy & Hiebert, 2011). These features have not been built into our simulations so far, but they will become important in later Workshops, where we simulate growing a vocabulary from scratch. The interesting point is the way that the simulations highlight these factors, and the way that they point us in the direction of a core vocabulary, even without the theoretical baggage that a linguistic account of a vocabulary brings with it.

The underlying question here seems to be how large is the basic core vocabulary. Program-9 suggests that the core needs to be fairly restricted, but not completely determined. Your experiments with the MaxN parameter will have shown that most models where the value of this parameter was set to a very low value (e.g., less than 20) produced a very different set of results from models where the MaxN parameter was constrained, but to a less strict value. However, this figure seems to be a lot smaller than the core vocabulary described by Carter.

The second question that arises out of Program-9 concerns the nature of the links between words in a model vocabulary network. At first glance, it seems obvious that the links are at some level the same as the kind of links that we find between words when we run word association tasks (cf. for example, Fitzpatrick & Thwaites, 2020). However, on closer inspection, this "obvious" connection is not as straightforward as it seems, and it does not work as well as we might expect. There are several reasons for this. Firstly, word association links are often symmetrical, with WordA acting as response to WordB and vice versa. It is not clear how lots of reciprocal links (WordA monitors WordB and vice-versa) would affect the performance of model vocabulary networks like the ones we are using in these Workshops. Secondly, word association links come in many different varieties. So, we have CAT~DOG, a traditional paradigmatic relationship, where a stimulus word evokes a response that it shares a number of semantic features with. Or we have CAT~FLAP, a syntagmatic association where the stimulus word evokes a response that completes a phrase with the stimulus word. And we have CAT~HAT, a straightforward phonological response. In contrast, the programs that we have used in this workshop have only a single type of link. Perhaps we should be modelling vocabularies which have several independent layers of connections, rather than a single layer of connections of a uniform type? In that case, the question morphs into a more complex discussion about how these different layers interact with each other. Thirdly, word associations come in different strengths: an association pair like MAN~WOMAN seems to be a lot stronger than an association pair like MAN~HAT, for example. Again, in contrast, the models we have been looking at so far treat all their connections as "equal", and it is not clear how the behaviour of the models would change if we allowed the strength of links to vary. Finally, as far as we can see, word association links are very numerous, with each word in a real lexicon connected to perhaps scores of other words. The models we have studied so far have only two connections between words, and it is not clear what the models would do if we allowed words to monitor more than two other words (see Wilks, Meara & Wolter, 2005 for some modelling work that deals with this question).

It would be possible to model all these features in our vocabulary networks, but it would be very difficult to do this without abandoning the simplicity of the approach we have adopted so far. In any case, we have already seen that even minimal structures like the ones we have worked with in these workshops give our model vocabularies some surprising emergent characteristics – plenty to be going on with for the moment. Our best guess is that word association links are not fundamental to vocabulary networks, but they may be emergent features – behaviours which arise because of other, more basic properties of the network. We will look again at this issue in a later Workshop.

Another reason for not wanting to identify word associations with the connections in our model vocabulary networks is that the only important feature of our own connections is that they identify the two watched words for each word in the network. However, the models do not care if WordX and WordY have semantic or phonological links with a target word, WordA. The only thing that matters to WordA is whether WordX and WordY are activated or not. An important consequence of this is that the connection patterns between words in a model vocabulary network do not have to be very specific. So, in order to activate WordA in a model network, it is not necessary for it to be linked to two specific inputs, say WordX and WordY. Any inputs would do, as long as they generate the same pattern of inputs as WordX and WordY. This has the effect of increasing the stability of a network that is in its attractor state, and making it less vulnerable to external perturbations. It also means that when one network – let us call it a child network – is learning to behave like another network – call this one a parent network – it is not necessary for the child network to learn the exact set of connections that characterises the parent network. Any equivalent set of links will work for the child network as long as they produce the same outputs. This makes the learning task very much easier for the child network than it appears to be at first sight. Again, we will explore this idea in more detail in a later workshop.

The third issue that arises naturally out of the simulation sets in Workshop 4 is the extent to which the behaviour of a vocabulary network might change depending on the network's overall state.

We saw in Program-7 that the number of ON words in a network makes a difference to the way a relinking event works. Networks where most words are ON tend to increase the number of ON words. Networks where most words are OFF tend to perform in the opposite direction. The interesting area lies between these two extremes – where the effects of swapping one watched word for another are much less predictable. This strongly suggests that we should not always expect an event to have the same effect. We might expect to find thresholds where the probability of a network suddenly jumping to a new attractor state increases dramatically. This is not so much of an issue with fixed size networks like the ones we have studied in this Workshop, but it might become important when we look at dynamic, growing networks.

Some Deeper Questions to Ponder

As usual, we end this Workshop with a set of deeper questions for you to ponder. There are no correct answers to these questions, but the work you have done with the

programs in this Workshop should be leading you to ask questions of this sort, and should be making you to think about how you might approach the questions by doing different kinds of simulation tasks.

- *How reasonable is it to assume that the connections between words are fairly stable?*
- *How would the models behave if one connection to a target word was fixed while the other was allowed to change?*
- *How would these models work if relinking events affected several words at a time, rather than just a single word?*
- *How likely is it that long chains of words that are dependent on each other will emerge in a model vocabulary network? How large would you expect these chains to be?*
- *Can you think of a way in which "better" connections between words might be modelled?*
- *Why might it be advantageous to have all the words in a network depending on a small core of words that provide them with input? Can you think of a mechanism which would allow a network of this sort to develop naturally?*
- *Suppose that you have two randomised models, and you wanted to ensure that they both operate in the same way, so that when WordX in the first model is ON so is WordX in the second model. Can this effect be achieved by restricted link-switching? Why might this be useful?*

Summing Up

If you have indeed pondered the deeper questions that we have listed at the end of each simulation set, then you will already be realising how working with simple simulations can get you thinking about some interesting theoretical issues which perhaps would not have arisen in the course of more conventional research. As always, we need to stress that the vocabulary network models that we have been working with are NOT intended to be realistic models of vocabularies. Rather they are simplified model networks, designed to help us understand how a network operates, how it differs from a mere pile of words, what its natural features might be, and what emergent properties a network of this sort might exhibit. The idea is that working with simplified models sharpens up our thinking, and helps us get a better grasp on the vague metaphors that commonly get used when we talk about vocabularies. We hope that you have found exploring these model vocabularies to be an interesting experience – one that has made you think about vocabularies in a different way. If that is so, then these basic simulations will have served their purpose.

However, we should point out that the simulations in this chapter are slightly unusual in that they do not actually model a specific process. Normally, when we work with simulations, we start off with a process that we want to model, build a simulation that we think captures the essential features of the process, and then examine how closely our model mirrors what actually happens in real life. Here we have been working in a different way. We started out with an interesting theoretical model, and then looked about for things that it might do: given a model vocabulary network, we asked how we could raise the activation in a network of this sort. That gave us the opportunity

to examine how various basic operations might affect a vocabulary network, but the models were designed to examine different types of events separately, rather than integrating them into a single model that could actually provide an explanation for some aspects of the behaviour of real vocabulary networks. In a sense, the features of the model were more important than the features of the real vocabularies that we were trying to model. As a way of getting you, the reader, familiar with the basic operations, this approach is fair enough, but it is rather more exploratory than we would really like. Our approach in these simulations was to examine some natural operations (turning words ON, changing the way words react to their inputs, and allowing changes in the links that connect words in a model vocabulary network), and to allow you to examine how these simple operations have unexpected outcomes. We hope that these unexpected results will have helped you to start thinking about vocabularies as networks, and to start critically evaluating the way the network metaphor is used in the vocabulary research literature. In the Workshops that follow, we will move on from this technical discussion: we will attempt to use the ideas we have developed in these early simulations to examine some real issues in some detail.

Despite the fact that our modelling so far has focussed on the mechanics of model vocabulary networks, four interesting ideas emerge from these preliminary simulations.

The most important one is methodological: simulating vocabulary acquisition by way of models is a really interesting way of making us think about the assumptions we normally bring to vocabulary research. Simulations have something of a bad press among Applied Linguists (cf. for example, Laufer 2005) on the grounds that they are not in close touch with "the real world". Obviously, a model vocabulary network is not intended to be an accurate reflection of a real vocabulary. Its purpose is to capture some essential features of how a real vocabulary might work, and to provide us with a tool which allows us to explore these features in ways which would be very difficult in the real world. Despite the deliberate simplifications involved, simulations can still be useful, especially in relation to the way we think about active/passive vocabulary and the dynamics of vocabulary growth.

We have also seen that the idea of maximising the number of activated words in a vocabulary network is not as straightforward as it appears at first sight. Although the distinction between active and passive vocabulary appears to be absolutely basic to the way vocabularies work, it is very clear from these simulations that it is a mistake to treat this feature as one which is intrinsic to individual words. Whether a specific word is ON or OFF in a particular network is not really a property of the word. Rather it is a property which derives from the way the specific word interacts with all the other words in the network, and one that is dependent on the entire structure of the network in which the word is embedded. The only way we can introduce a permanent uplift in the number of active words is by changing the overall structure of the vocabulary network, so that it has a different attractor state. This is a pretty impressive outcome for a set of very simple simulations: already we are being forced to ask some deep questions about things that had previously seemed obvious. What does actually happen when a word we know moves from Passive to Active status? Does the idea of an Active/Passive continuum really stack up? Is it beneficial for a vocabulary network to contain a lot of activated words? Is there a natural relationship between the number of active words and the overall size of a vocabulary network? In our experience, working with

simulations almost always generates deep questions of this sort, and this is one of the reasons why this type of work is so exhilarating.

Thirdly, the simulations in the Workshops so far have provided some hints that the dynamics of vocabulary growth might be much more important than we originally thought. These simulations have all used fixed size models, where the vocabulary network contained 1000 words, and where most of their basic properties were fixed in advance. However, as we worked with these models, it has become increasingly clear that some of these fundamental properties might not be fixed forever. Indeed, the most effective ways of increasing the number of active words in our models involved changes to the structure of the network (relinking events) or changes to the way words respond to their watched words. These are both fundamental properties of words in a model vocabulary network, whereas the current state of a given word in a network appears to be a transient feature, rather than a fundamental one. We also found that some events generated different types of behaviour in our models depending on the overall state that a vocabulary network finds itself in. Crudely, vocabulary networks that have a lot of activated words behave very differently from networks where the number of activated words is low. And vocabulary networks which have about the same number of words that are easy to activate and words that are less easy to activate behave very differently from networks where one type of word predominates. All this suggests that the initial working assumptions which we built into out model networks are missing some important characteristics of networks that grow and change. We will follow this idea up in a later Workshop, where we will build vocabulary networks from scratch, rather than presenting them as a *fait accompli*, fully formed and fully functional when they are first initialised.

Finally, the last idea to emerge from these early simulations arises from the fact that we have been able to model some interesting features of vocabulary networks without any recourse to semantics or a specifically lexical structure. This does not mean that "lexical structures" are not important to a vocabulary, but it IS surprising that we can get so many interesting effects without having to invoke a specific lexical structure – most of the effects that we have noted in this chapter are simple properties of networks, rather than specific properties of vocabulary networks. At the very least this ought to make us ask whether we really need to invoke a separate lexical structure in order to explain the basic behaviour of a vocabulary network – what additional features does a lexical structure bring to a network that are not there already embedded in simpler structures? What behaviours in a real lexicon would **require** us to invoke a specifically lexical structure? And how would a specifically lexical network structure differ from the simpler generic networks that we have been looking at so far?

Again, we will look at these issues in more detail in later workshops, and in the next two workshops will begin to use the framework that we have developed so far to look at some real issues in L2 vocabulary research, beginning with the question of attrition and vocabulary loss. This area is an important, but under-researched aspect of the way vocabularies work, and the reported research findings are very contradictory. Using models to investigate vocabulary loss throws some light on these contradictions, and raises some important real-world questions about how vocabulary loss can be assessed.

References

Carter, R. (1987). Is there a core vocabulary? Some implications for language teaching. *Applied Linguistics, 8*(2), 178–193.

Fitzpatrick, T., & Thwaites, P. (2020). Word association research and the L2 lexicon. *Language Teaching, 53*(3), 237–274.

Krashen, S.D. (1982). *Principles and practice in second language acquisition*. Pergamon Press.

Laufer, B. (2005) Lexical frequency profiles: from Monte Carlo to the Real World. A response to Meara. *Applied Linguistics, 26*(4), 581–587.

Nagy, W.E., & Hiebert, E.H. (2011). Toward a theory of word selection. In M.L. Kamil, P.D. Pearson, E.B. Moje, & P.P. Afflerbach (Eds.), *Handbook of reading research* (Vol. 4, pp. 388–404). New York: Longman.

Wilks, C., Meara, P.M. & Wolter, B. (2005). A further note on simulating word association behaviour in a second language. *Second Language Research, 21*(4), 359–372.