OPEN ACCESS

Vocabulary Learning and Instruction
ISSN 2981-9954
Volume 14, Number 1 (2025)
https://doi.org/10.29140/vli.v14n1.2139

Castledown

Vocabulary Networks Workshop 5 Modelling Attrition in a Vocabulary Network

Paul Meara
Swansea University and University of Oxford

Imma Miralpeix
Universitat de Barcelona

Abstract

This paper is part 5 of a series of workshops that examines the properties of some simple models of vocabulary networks. While previous workshops dealt with activating words in the network, this workshop focuses on vocabulary loss. We will simulate two possible ways of modelling attrition: (a) explicitly turning active words OFF, and (b) raising the activation threshold of a few words in a network. The workshop is linked to an online practice room where readers can explore these processes for themselves.

Keywords: vocabulary networks, lexical attrition, vocabulary loss, L2 vocabulary.

Introduction

The standard way of examining vocabulary attrition in speakers of a second language (L2) is basically very straightforward. We take a group of L2 speakers who have not used their language for some time and give them a vocabulary test, usually a test of their ability to translate items from their first language (L1) into the L2. We then use the results of this test to make inferences about their remaining vocabulary competence in the L2. Repeating the test many times allows us to plot the course of vocabulary loss over a period of months, or even exceptionally a period of years. The data then allow us to make inferences about the long-term pattern of vocabulary loss in L2 speakers.

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

A typical example of this kind of approach is to be found in Weltens et al. (1986). An alternative approach, found in Bahrick (1984a,b), is to compare groups of L2 speakers in a cross-sectional study. This approach allows us to make inferences about vocabulary loss in individual L2 speakers based on the residual vocabulary in the different groups. The main patterns of attrition reported in the literature are summarised in Weltens and Grendel (1993).

Readers of these workshops will immediately recognise that there is a problem with these approaches: effectively, they all treat a vocabulary as an unstructured collection of words rather than as a structured network (a point that is extensively discussed in Larson-Hall, 2019). The research sees attrition as a process that affects individual words, but the knock-on effects for other words are rarely considered. In this workshop we will look at some features of how lexical attrition might work in the simple network models that we have been using so far.

These models – technically known as k = 2 random autonomous Boolean networks (Kauffman, 1993) – are extremely simple, and are not intended to realistically represent how actual vocabularies function. Their main purpose is to encourage you to think about vocabularies as networks, and to explore what properties a lexical network might exhibit. So far, we have worked with networks consisting of 1000 words, where each of these words has three main properties: (a) each word has an activity state, which can be ON or OFF. These states are initially assigned at random; (b) each word in the network is randomly linked to two other words in the network. These links are also assigned at random; (c) every word monitors the words that it is linked with and responds to the activity state of these watched words by activating or deactivating itself. Each word has a "Boolean Type" – a simple rule which determines how it reacts to the two words it monitors. Some words are AND words: they will become activated if both of their watched words are active. Some words are OR words: they will become activated if both or only one of their watched words is activated. These Boolean types are assigned randomly when the network is set up. In the previous workshops we saw that this very simple structure has some surprisingly complicated properties. The most important property is that a network of this sort is much more stable than you might expect. These networks will usually settle into an attractor state where some words are normally ON (active vocabulary), and others are normally OFF but can be turned ON by events that influence the vocabulary network (passive vocabulary). We also sometimes find a small number of words that flip between these two states, sometimes ON and sometimes OFF (unstable vocabulary).

In this workshop we will be using the Boolean network models to examine some interesting features of vocabulary attrition.

Modelling Attrition

Reducing the Activity in the Network

We can simulate attrition in a model vocabulary using the tools that we have already used in the previous workshops. The obvious approach is to ask if we can model attrition using the simple expedient of deactivating words by putting them into their OFF state. This approach is similar to the approach that we used in Workshop 2, where we attempted to increase the overall activity in a model vocabulary network by turning words ON. Here, we want to find a plausible way of **reducing** the overall activity in a network, and at first sight, it looks as though the easiest way to achieve this aim would simply involve turning words OFF. We do this using Program-11, which you will find on the Workshop Home Page: https://www.lognostics.co.uk/Workshop The interface for Program-11 is shown in Figure 1.

This interface will be familiar to you from the earlier workshops. You have four parameters that you can experiment with. The **NTWK** parameter determines the basic structure of your network and the characteristic features of the words in the network. In this simulation set, each event turns some words OFF. How many words get turned OFF at each event is set by the **sEv** parameter. The **nEv** parameter determines the number of events that will be implemented in the course of your simulation. Finally, the **rEV** parameter determines which words are affected by each event. These parameters allow you to turn OFF a small number of words and explore how doing this impacts on the overall level of activity in a model vocabulary network.

As usual, before you start using the program, you should think about how you would expect it to behave, and what features might be important in determining the outcomes it will give you. We know that networks like the ones we studied in Workshops 2-4 usually move towards an attractor state, and that these attractor states are surprisingly stable. We also saw that nudging a network out of its attractor state by turning words ON was not an effective way of permanently increasing activity in the network. In most cases, the networks resisted being nudged in this way: turning a small number of words ON leads to a brief, transient rise in the number of active words in the network, but these spikes disappear as soon as we stop turning words ON, and the network returns very quickly to its attractor state. Even when we turned very large numbers of words ON, it was unusual to find a permanent shift in the level of activity in a network. This leads us to ask whether model vocabulary networks would react in a similar way when words are turned OFF: will a network return to its attractor state

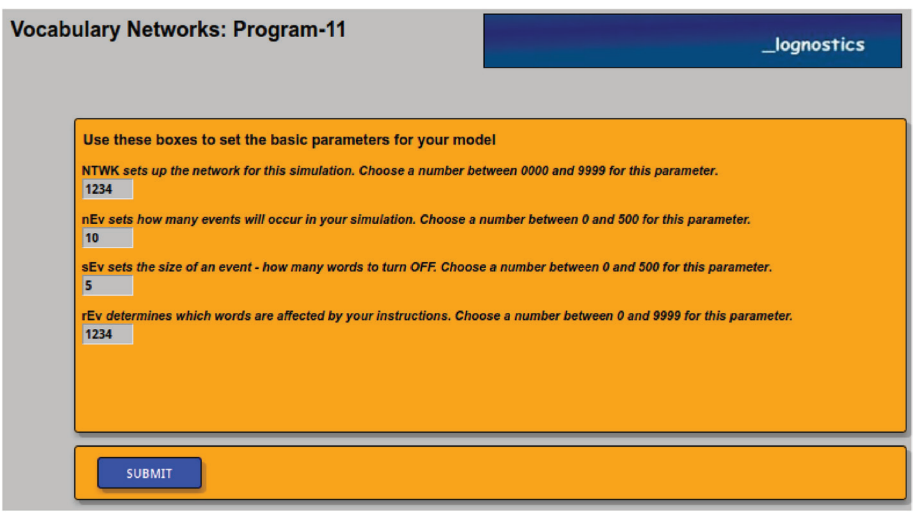

Figure 1 *Program-11: The Data Input Screen.*

after we turn a small number of words OFF? And if we turn very large numbers of words OFF, will we find a permanent shift in the activity level? Will a network repair itself in this case? How much damage can a network absorb and still repair itself?

To begin, you can set the nEv parameter and the sEV parameters to 0. This will give you a network where no events take place, but they will allow you to use the NTWK parameter to find a few examples of networks that have a high number of activated words (e.g., over 800) in their attractor state. Make a note of the relevant NTWK values, as we will be working with these models in the rest of this workshop.

In our first attempt at simulating attrition in a model vocabulary the question we are asking is **how a model vocabulary network will react to having some of its words deactivated**. Would we expect this to have a lasting effect on the overall activation level of a network? Probably not: we saw in Workshop 2 that turning words ON did not make much of a contribution to the overall level of activation in a network, and we might expect that turning words OFF would likewise not result in a permanent loss of activation. On the other hand, we cannot assume that turning words OFF is just the mirror image of turning words ON, and it is reasonable to speculate that turning OFF a large number of words might result in a permanent loss of activation. The question then becomes: how large a number of deactivated words can a network recover from? Program-11 lets you explore questions of this sort.

From the setup screen, set the NTWK parameter to one of the values that gives you a network with a high number of activated words. Set the sEv parameter to 5 and set the nEv parameter to 10. Choose a random number for the rEv parameter. These settings will give you a simulation where you have a vocabulary of 1000 words, most of which are activated. Your simulation will run through 1000 updates of the model. Ten attrition events will take place, and each of these attrition events will turn five words OFF. The results of this simulation will look something like Figure 2.

In this example (NTWK = 3500; nEv = 10; sEv = 5; rEv = 1234), the network finds an attractor state where 880 words are active. Attrition events take place at updates marked by red dots at the bottom of the display. As you might have predicted, these events have only a marginal effect on the overall activation level of this network. Most events have no noticeable effect at all – words that are turned OFF are immediately restored to their active state. Where two or three events take place in quick succession we find a small ripple in the activation level, but these ripples are short-lived.

The obvious question to ask next is whether we can get a lasting decline in the number of activated words if we make the events larger. You can explore this idea with your own models by changing the sEv parameter to a higher value. Figure 3 shows an example of what happens when we raise this value to 50, while keeping the number of events to 10.

You should find that the network still returns to its attractor state after each event. The ripples are more persistent, but they still fail to produce a permanent fall in the overall number of active words. You can experiment with larger values for the sEv parameter, but even when hundreds of words are turned OFF in each event, you should find that the network will normally find its way back to its attractor state.

Now restore the value of sEv to 5 again but raise the value of the nEv parameter to 100. This will let you run simulations where the number of events is 100 and each

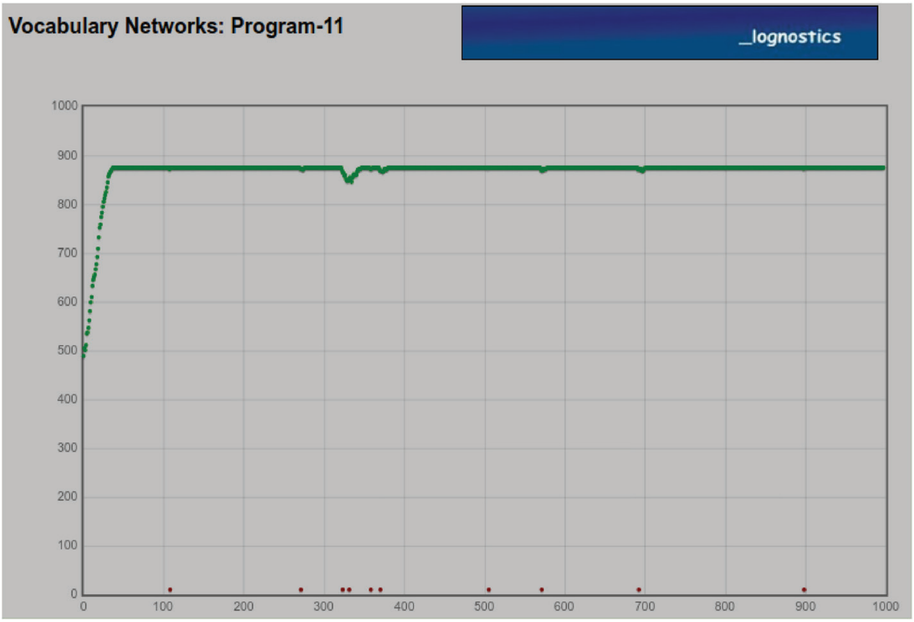

Figure 2 *NTWK = 3500; nEv = 10; sEv = 5; rEV = 1234.*

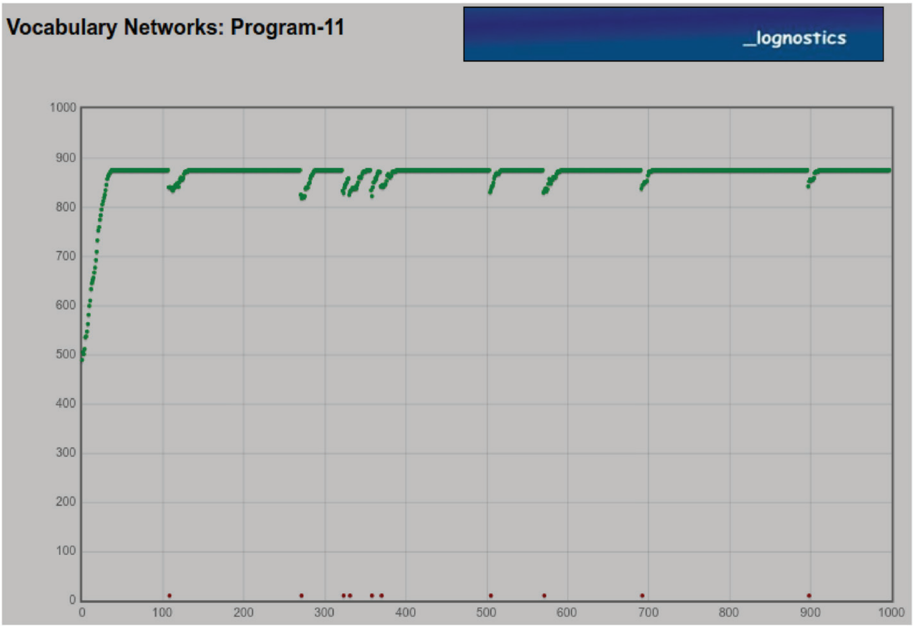

Figure 3 *NTWK = 3500; nEv = 10; sEv = 50; rEV = 1234.*

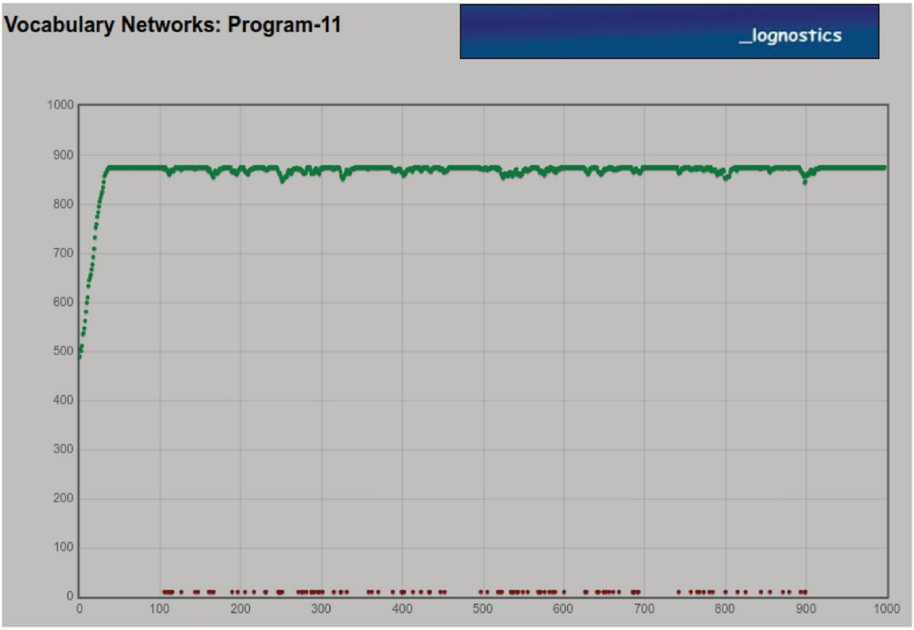

Figure 4 *NTWK = 3500; nEv = 100; sEv = 5; rEv = 1234.*

event turns 5 words OFF. Figure 4 shows you how Network 3500 reacts to this set of parameters. Once again, we get a number of ripples in the overall activation level of Network 3500, but when the events stop at update 900, we soon find that the network returns to its attractor state with 880 words ON.

Figure 5 shows how Network 3500 reacts to much higher values of sEv and nEv. Here we have 200 events, each of which turns OFF 100 words. The resulting graph shows some very large swings in the number of activated words, with an all-time low around update 530, where the number of activated words has fallen to 350. But even with these extreme values, the network returns to its attractor state once we stop turning words OFF at update 900.

You should use Program-11 to run your own simulations and explore these ideas further. We have put some suggestions for you to follow up in the list below:

- *Are networks with only a small number of ON words more stable than networks with a larger number of ON words?*
- *Do these results generalise to models with a low level of activation?*
- *Are networks with a smaller number of ON words slower to recover from an event?*
- *What role does parameter rEV play in these simulations? (This parameter controls the specific words that are turned OFF by an event).*
- *If you vary the value of the NTWK parameter, you will get a different underlying network. Are some networks more resilient than others? Can you identify any networks that fail to return to their initial attractor state?*

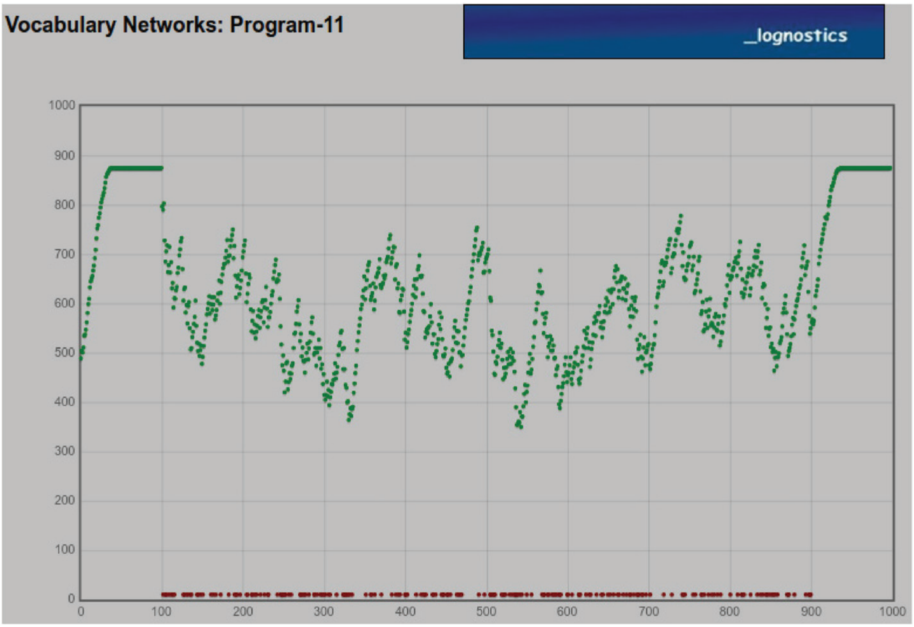

Figure 5 *NTWK = 3500; nEv = 200; sEv = 100; rEv = 1234.*

You should also be thinking about some bigger questions:

- *Will the models we have looked at here scale up to a larger vocabulary size?*
- *Why do the models restore themselves to their attractor state?*
- *Is it realistic to explore models where the number of active items is very high?*
- *The models we have looked at so far have all had random connections between their words. How might a non-random network be affected by having its words turned OFF?*
- *What sort of mechanism might turn words OFF in real life?*
- *What sort of mechanism would lead to frequent and large loss events in real life?*

On balance, these results confirm our suspicion that turning words OFF is not an effective method to model vocabulary loss. It IS possible to reduce the number of active words in a network, but only if we continuously suppress active words. However, it is difficult to think of a plausible mechanism that would allow large numbers of words to be continually suppressed: active suppression of words just does not feel like a cost-effective way of maintaining a vocabulary. You would not want to argue, for example, that an L2 learner cannot remember the German vocabulary they learned in school because it is being continuously suppressed. If that were the case, we would expect to find occasional instances where the suppression mechanism stops working, and cases of spontaneous recovery of an L2 appear. There are a few cases of this sort in the literature, but they mostly appear in clinical research, and they do not seem to reflect what happens in ordinary circumstances (see **https://www.sciencealert.com/people-keep-waking-up-from-head-injuries-speaking-a-different-language** for some interesting anecdotal examples).

On the other hand, it does make sense to argue that some vocabulary in an L2 will stay with you even if you stop using it for a long time. Your school German vocabulary will gradually revert to an attractor state if you stop using it, but it is very unlikely that ALL the words in your German vocabulary will disappear. This idea has been explored in a series of papers by Bahrick (e.g., Bahrick, 1984a; Bahrick et al., 1993, 1994), and it is easy to see how Bahrick's idea of a "permastore" of words that do not undergo attrition arises naturally in models where attractor states are important. In a later section of this workshop, we will look at alternative ways of modelling attrition in a model network, but before we do that, it is worth examining catastrophic loss in a vocabulary network (i.e., what happens when we reduce the network activity to zero by turning all its words OFF).

Catastrophic loss

Program-12 (**https://www.lognostics.co.uk/Workshop/)** simulates a catastrophic loss of vocabulary, turning ALL the words in a network OFF. This puts the network into an attractor state where no words are active. You can check this out by choosing any value for the NTWK parameter (e.g., choose one that gives you a high level of activity in the network's attractor state). Set the value of the nEv parameter and the sEv parameter to 0. When you run Program-12 with these values, you will find that the network settles into an attractor state at around update 50. The program is set to turn all words OFF in a single event at update 100, and the green line that shows the number of active words in the vocabulary will fall to zero at this point.

Normally, a model vocabulary network with all its words turned OFF will remain in that state. But the question to ask now is what happens when we inject a small amount of activity into the network by allowing events to turn a few words back ON; that is, after all words are turned OFF, we can model some recovery events. You can use Program-12 to do this by setting the nEv parameter to 5 and the sEv parameter to 10. This will give five small kicks to your deactivated network; at each kick, 10 words are turned ON again.

Figure 6 shows one example of this combination of parameters. NTWK 3500 settles into its initial attractor state where about 880 words are ON. The catastrophe event happens at update 100, and the activity level immediately drops to zero. After that, five recovery events take place at updates 275, 350, 360, 690 and 900. At the first recovery event, ten words are reactivated, and this sets up a ripple of activation which causes the network to move into a new attractor state where about 190 words have been reactivated. The second and third reactivation events take place shortly after this, leading to a massive increase in the number of active words. The network eventually settles into a new attractor state where the number of active words is just a little short of the initial attractor state. The fourth recovery event takes place just before update 700, and fully restores the network to its initial attractor state.

This recovery pattern is quite surprising. Let us imagine that what we are modelling here is a patient in a clinical setting. Our patient has completely lost their vocabulary

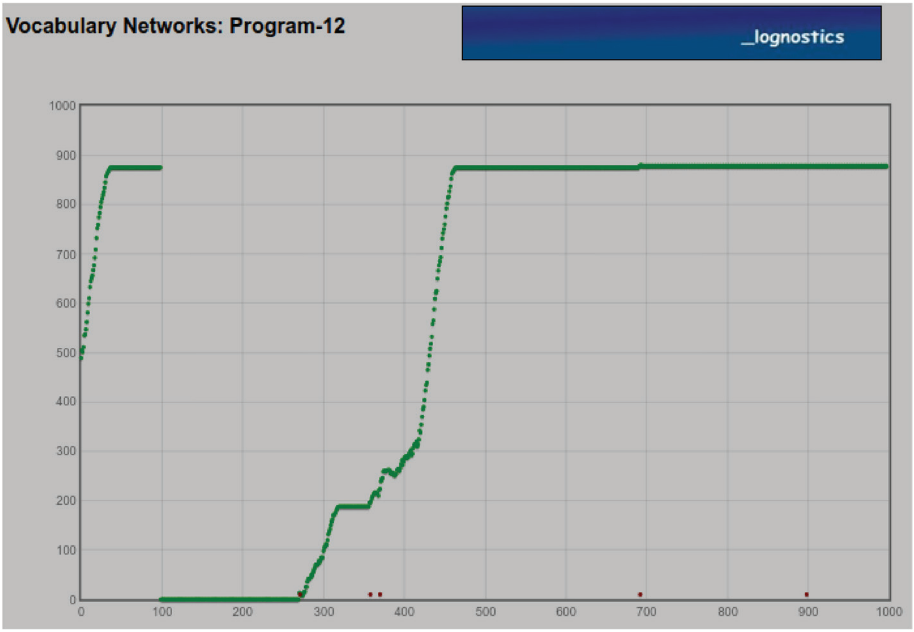

Figure 6 *NTWK = 3500; nEv = 5; sEv = 10; rEv = 1234.*

at update 100, perhaps because of a massive stroke event, but after a tiny intervention (turning ON a mere 30 words) our patient's vocabulary recovers almost completely. The model implies that when a catastrophic loss event takes place, the model vocabulary retains a memory of what its structure looks like, and with a little help it can reconstruct its original shape.

Not all models recover in this way, and you can use Program-12 to explore some of the different ways that recovery manifests itself. Some suggestions to help your explorations are listed below:

- *Make a list of twenty networks with different levels of activity in the initial attractor state.*
- *Set the value of nEv to 5 and set the value of sEv to 10. This combination will give you five recovery events, each of which temporarily re-activates 10 words.*
- *Choose a random value for the rEv parameter. Now run each of your NTWK values with these parameters.*
  - *How many of your models recover fully from the catastrophic event?*
  - *How many of your models partially recover?*
  - *How many of your models do not show any long-term recovery?*
  - *How many different recovery patterns can you identify?*
- *Change the value of the rEv parameter by choosing a different random value. Does this change the recovery patterns?*
- *Why are some values of rEv effective, but others are not?*

- *How might recovery patterns differ if the connections between words in a network were non-random?*
- *Does recovery depend on events occurring in a quick sequence?*
- *How do the values of nEv and sEv interact?*
- *Do networks with a low number of active words in their initial attractor state show different recovery patterns?*
- *Will these results generalise to much larger models, say, a model with 10,000 words?*
- *Will these results generalise to much smaller models, say, a model with only 50 words?*

As usual, there are no right answers to these questions: their purpose is to make you realise that networks behave in peculiar ways and have surprising emergent properties.

Raising the Activity Threshold of Words

In this section, we look at another way of simulating vocabulary loss, which may be a more effective way to model attrition.

In Workshop 3 we noted that the overall level of activity in a network was very strongly affected by the number of words in the network that are easily activated. Basically, lowering the activity threshold of a small number of words massively increased the number of ON words in the network. The question we ask in this section is whether a similar process could account for vocabulary loss. We will examine how raising the activity threshold of words affects the overall level of activity in the network.

The main algorithm for these simulations is summarised in Figure 7. Note that this algorithm appears to be much simpler than most of the events that we modelled in our earlier workshops. The algorithm randomly selects a target word and changes the way it reacts to the two words it monitors (its Boolean Type), but as we shall see, this process is more complicated than it looks to be at first sight.

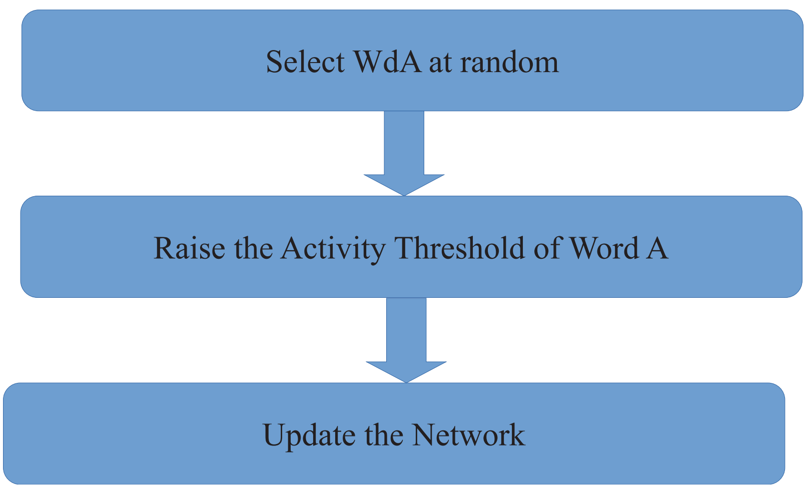

Figure 7 *The Structure of an Attrition Event.*

To understand how the algorithm works, you need to know how the information about individual words is stored in the models that we have used so far. Each word in the vocabulary is stored as a vector – a series of numbers that encode the important characteristics of the word. Table 1 below gives you an example of a vector for a single word. The model vocabulary is made up of 1000 vectors of this type when the vocabulary is initialised. ID identifies the word; L1 and L2 identify the two words that are watched by the word; CV determines the current state of the word (ON or OFF). The value of the BT code determines whether the word is easily activated (OR words require only one of their watched words) or more difficult to activate (AND words only become activated if both of their watched words are active). Apart from the ID code, all these features are randomly assigned when the network is initialised. Table 1 shows you that Word 27 is an OR word that watches Word 245 and Word 356. It goes ON if at least one of these watched words is active.

Step 2 of the algorithm can now be interpreted in different ways. The obvious interpretation is **identify an OR word and raise its BT value to 2.** This interpretation is straightforward, but it will become increasingly difficult for the program to find an OR word as the number of AND words increases. A slightly more complicated interpretation is **choose a random word; if the chosen word is an OR word then raise its BT value to 2; if the chosen word is already an AND word, then do nothing.** This interpretation of the instruction means that events will become increasingly ineffective as the number of AND words grows. The interpretation that we will adopt here is a third option: **choose a random word and raise its BT value by 1**. This interpretation is elegantly simple, but it has some interesting knock-on effects, discussed in more detail below.

Now, let us consider how this third interpretation works out in practice. There are several conditions to consider:

a) ***the selected word is an OR word, but none of its watched words is ON.*** In this condition, the selected word will be OFF when the attrition event takes place. Its activity threshold will be raised to 2 and the word remains OFF. (No change to the overall activity level)
b) ***the selected word is an OR word, and one of its watched words is ON.*** In this condition, the selected word will be ON when the attrition event takes place. Its activity threshold will be raised to 2, but it only has one active watched word, so it will turn OFF. Note that this loss may have knock-on effects for subsequent updates. (Overt vocabulary loss)
c) ***the selected word is an OR word, and both of its watched words are ON.*** In this condition, the selected word will be ON when the attrition event takes place. Its activity threshold will be raised to 2, but since both its watched words are ON, the selected word will remain ON. (No change to the overall activity level)

Table 1 *Example of a Vector for Word 27.*

Characteristics	ID	L1	L2	BT	CV
Example	27	245	356	1	1

d) ***the selected word is an AND word but none of its watched word is ON.*** In this condition, the selected word will be OFF when the attrition event takes place. Its activity threshold will be raised to 3, and the word will remain OFF. (No change to the overall activity level)
e) ***the selected word is an AND word, and one of its watched words is ON.*** In this condition, the selected word will be OFF when the attrition event takes place. Its activity threshold will be raised to 3 and the word will remain OFF. (No change to the overall activity level)
f) ***the selected word is an AND word and both of its watched words are ON.*** In this condition, the selected word will be ON when the attrition event takes place. Its activity threshold will be raised to 3 and the word will turn OFF. Note that this loss may have knock-on effects for subsequent updates. (Overt vocabulary loss)

It is worth pointing out that only two of these scenarios always result in loss of activity in the network: condition *b* and condition *f*. In the other four conditions, there is no immediate change in the overall activity level of the network, though. Conditions where the selected word is an OR word will always result in an increment in the number of AND words.

The conditions involving AND words (whose activity threshold is 2 when the attrition event takes place) are interesting for another reason. When an AND word is affected by an attrition event, its activation threshold rises from 2 to 3. But since the words in these simple networks only monitor two other words, it is impossible for this new threshold condition to be met, and the target word will never be able to reactivate itself. In effect, what we have here is a new class of words that we will call '**zombie words'.** These words will be discussed in more detail later.

You can examine the effects of this type of attrition event using Program-13, which you will find on the Workshop Home Page: **https://www.lognostics.co.uk/Workshop/index.htm.**

The data input screen for this program gives you three parameters that you can experiment with. As usual, **NTWK** sets up a 1000-word model vocabulary; **nEv** sets the number of events in your simulation; **rEv** determines which words are affected by the attrition events. In this condition, each event affects only one word, so there is no sEv parameter.

As always, before you start to work with Program-13 you need to think about how you would expect a network to respond to attrition events that raise the activity threshold of a single word. Will vocabulary loss be gradual or sudden? How quickly will attrition make itself apparent? Will the attrition events be affected by the number of activated words in the network words thresholds? Will all attrition events result in overt vocabulary loss? How much variation would you expect to find in the way networks respond to repeated attrition events?

We will begin by looking at how a network with many activated words reacts to many attrition events. You should have a list of NTWK codes that give you models where the number of active words in the vocabulary is over 800. Choose one of these values. Set the value of the nEv parameter to 150 and choose a random number for the value of the rEv parameter.

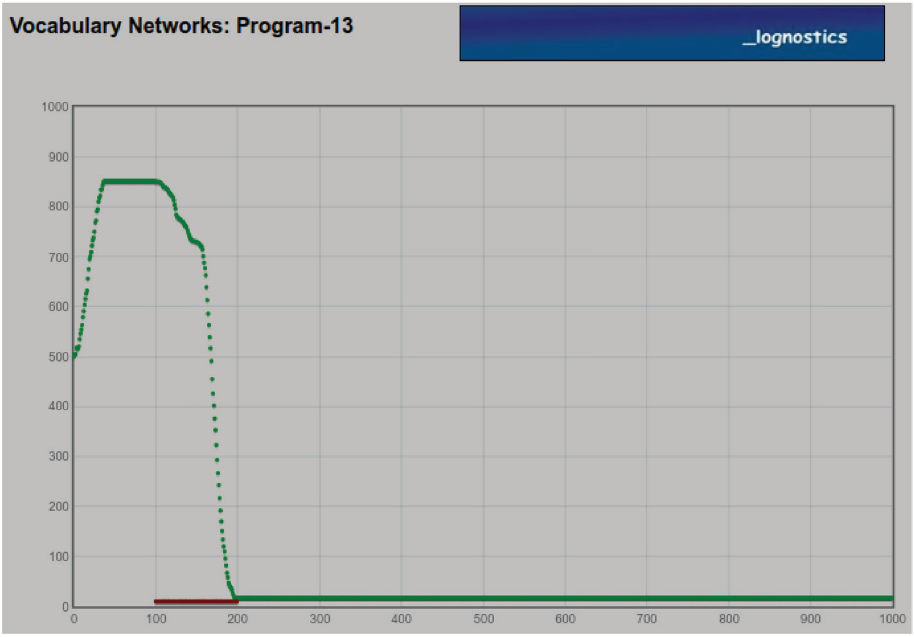

Figure 8 *NTWK = 3029; nEv = 100; rEv = 1234.*

Figure 8 shows you how model 3029 reacts to 100 attrition events where each event raises the activation threshold of one word. Each event happens in quick succession. Your own simulations will look somewhat different from this, depending on the value you choose for the NTWK parameter.

This model vocabulary finds its initial attractor state with 859 words ON. The first attrition events start at update 100 and one word is degraded at each of the next 100 updates (indicated by the red line at the bottom of the graph). Vocabulary loss appears immediately, with a steep reduction in the overall activity level in the network. However, around update 150, we get a sudden catastrophic collapse in the network, and by update 200, only a handful of words remain activated. This is quite a spectacular result. On the face of it, the network has collapsed after fewer than one in ten of its words have been degraded.

However, we could argue that the collapse is caused by a much smaller number of events than Figure 8 suggests. As we saw earlier, not all attrition events cause an immediate loss in the total active vocabulary. To exemplify this, we need to consider the six conditions listed in this section (See Table 1). Table 2 shows the number of words in each of the six conditions at the point where the attrition events start at update 100.

We have seen above that only condition *b* and condition *f* result in immediate vocabulary loss, so when the attrition events start at update 100, the chance of an immediate vocabulary loss resulting from a random attrition event is only 483 (125 in *b* + 358 in *f*) in 1000 – slightly less than one event in two. Note, though, that these proportions are not fixed: they will change as the simulation develops.

Table 2 *Number of Words in Each Condition When Attrition Events Start at Update 100.*

The Six Condition Types	T100
a) ***the selected word is an OR word but none of its watched word is ON.***	10
b) ***the selected word is an OR word and one of its watched words is ON.***	125
c) ***the selected word is an OR word and both of its watched words are ON.***	367
d) ***the selected word is an AND word but none of its watched word is ON.***	12
e) ***the selected word is an AND word, and one of its watched words is ON.***	127
f) ***the selected word is an AND word and both of its watched words are ON.***	358

The pattern of vocabulary loss that we find in Figure 8 – a short period of stability followed by a catastrophic collapse in the overall activity level of the network – seems to be the norm in this simulation set, but it is not the only kind of pattern that we find in these simulations. You should experiment with your own models and note how many different patterns of attrition you find. The questions below will guide your explorations:

- *Do all your models undergo a total collapse in their activity levels? (change the value of the NTWK parameter to explore this idea).*
- *What is the average number of attrition events needed for a total collapse?*
- *What is the largest number of events you get where no overt vocabulary loss occurs?*
- *Are models with a high level of activity more vulnerable to total collapse than vocabularies with a lower level of activation?*
- *How many events are* **necessary** *for a collapse to appear?*
- *What is the average length of a period of stability in these models?*
- *Can you predict when a catastrophic collapse is likely to occur?*
- *How many different patterns of vocabulary loss can you identify in your models?*
- *Will these models scale up? (e.g., would a 5000-word vocabulary perform like the 1000-word vocabularies used in these simulations? What about a 20000-word vocabulary)*
- *How important is the value of the rEv parameter in these simulations? (What does this tell you about the individual events?)*
- *Would the models show different characteristics if they were non-random? e.g., if an attrition event was more likely to target AND words than OR words, or if the words monitored by other words all come from a small core that resists attrition?*
- *Can you design a small network (5 words or so) where all the words are resistant to attrition? What is the probability that a resistant cluster of nodes would arise by chance in a larger network?*
- *Would your models return to their initial attractor state if you were able to turn a few words back ON?*
- *In Program-12, an attrition event takes place before each update. How would the models react if attrition events were spaced out and happened infrequently?*

As usual, there are no correct answers to these questions, but they should be making you think about the assumptions we usually bring to vocabulary research, and to vocabulary attrition.

Program-14 is an attempt to answer the last of these questions. In Program-13, all the updates occurred together in a continuous bunch. However, it seems more probable that attrition events occur in real life at random intervals. Program-14 gives you some control over these random events by adding an additional parameter, **mEv**: a low mEV value clumps events towards the start of the simulation, while a high value spreads out the attrition events. Obviously, when the events are spaced out rather than clustered together at the start of the simulation, we would expect the model vocabularies to appear to be more resistant to attrition events. You can see this in Figure 9.

Figure 9 illustrates a network (NTWK = 6002) which settles into an initial attractor state where 865 words are ON. The network experiences 25 attrition events spread over 1500 network updates. Most of these updates have no immediate effect on the overall activity level of this network, but we do find two large vocabulary losses, the first beginning at update 530, and the second beginning at update 720. Only five other vocabulary loss events appear in the data, and these are very small.

It is difficult to generalise from this example. Random attrition events seem to generate a huge amount of variation in the vocabulary loss events recorded in our models. You should explore this idea for yourself. Choose a set of 20 values for the NTWK parameter, set the nEv parameter to 25, the mEv parameter to 150, and choose a random number for the value of the rEv parameter. Then ask the questions listed below:

- *How many of your models have a high activity level in their initial attractor state ('high' means more than 800 ON words). How many have a low value (fewer than 250 ON words)?*

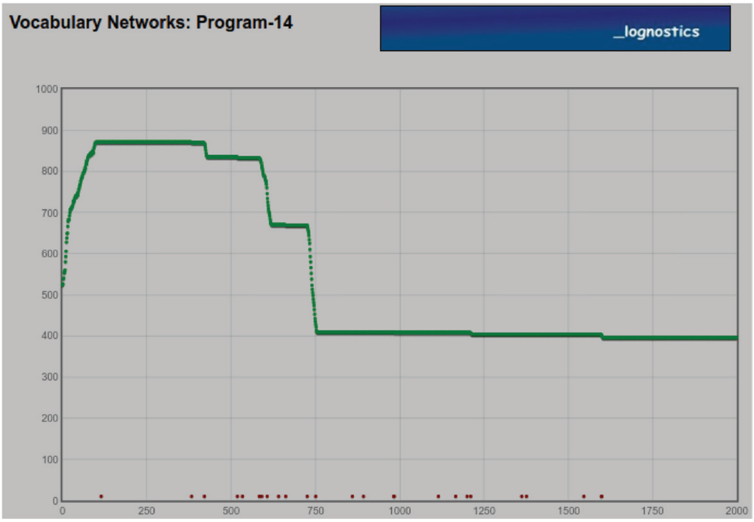

Figure 9 *NTWK = 6002; nEv = 25; mEv = 1500; rEv = 1234.*

- *With your chosen parameters, do you always find at least one large fall in the number of ON words in the network?*
- *How large is the biggest fall in your networks?*
- *How many attrition events are necessary for a large fall to emerge?*
- *How many attrition events are sufficient for a large fall to emerge?*
- *These models seem to have long periods of stability punctuated with short periods of vocabulary loss. How many loss events do you typically get with these parameter settings? How long is a typical period of stability?*
- *Are stable periods shorter at the beginning of the simulation?*
- *Typically, how many ON words does a network have after 2000 updates?*
- *How much variation is there around this figure?*
- *Set the mEv parameter to 50 and the nEv parameter to 100. Now run a set of simulations where you gradually increase the value of mEv. How does this affect the pattern of attrition in your models?*
- *How would you interpret the data shown in Figure 10?*

Discussion

Several important ideas emerge from the simulations in this workshop.

Firstly, the simulations suggest that we need to make a distinction between an **attrition event** and a **vocabulary loss event**. Both Program-13 and Program-14 illustrate how an attrition event (raising the activity threshold of one word) does not always result in a loss in the overall activity level in a vocabulary network. The data you

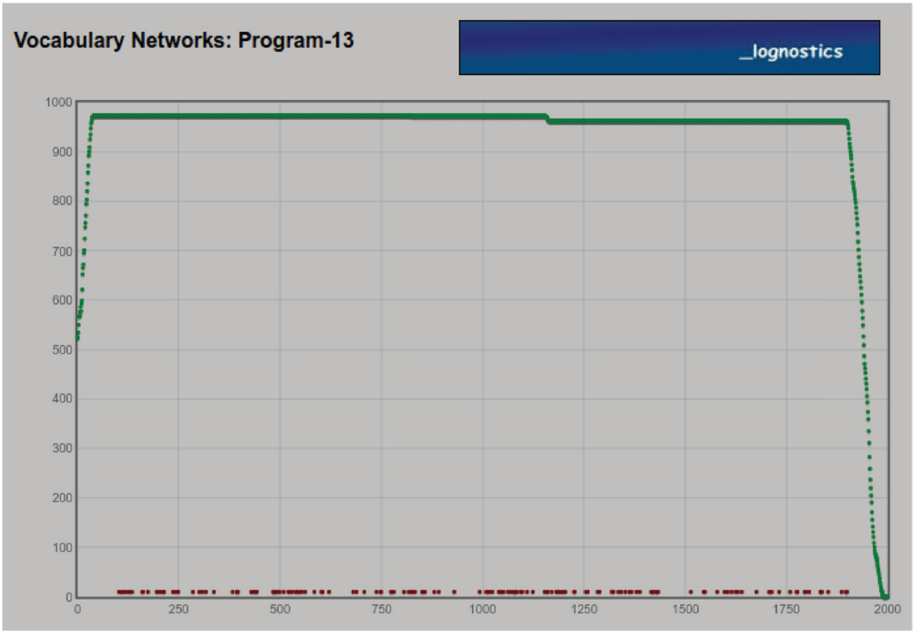

Figure 10 *NTWK = 3001; nEv = 150; rEv = 1234.*

collected from Program-14 should have shown you that these networks can sometimes be very resilient, sometimes absorbing many attrition events before they show any overt signs of vocabulary loss. This is an important idea, because much of the research on attrition focusses on vocabulary loss rather than attrition events. For example, in a typical L2 attrition study we might test an L2 learner's knowledge of L2 vocabulary using a standard vocabulary size test such as Nation and Beglar's VST (Nation & Beglar, 2007), and then retest them later, say six months after the original test. A lower score on the second test would be taken as a sign that attrition has taken place. However, as we have seen, attrition events can take place without any sign of overt vocabulary loss, so it is quite likely that a standard vocabulary test will fail to pick up that attrition events are happening. We will explore this idea further in the next workshop.

The second important idea is that a **catastrophic collapse** in the number of active words is not an unusual occurrence in these models. With Program-13, where the attrition events take place in quick succession, catastrophic collapse takes place after a surprisingly small number of attrition events have been implemented. With randomly spaced attrition events (Program-14) a single catastrophic collapse appears to be a less likely outcome: instead, a small number of medium size vocabulary loss events seem to be responsible for most of the decline in activity. It is not immediately obvious why this difference should appear in the data. Our best guess is that an attrition event will sometimes generate a ripple in the number of words being turned OFF as a result of a single word changing its status. Program-14 suggests that the spacing of the events might allow a network to partially recover from these effects as long as they are infrequent. You should also note that the occurrence of a large vocabulary loss event, especially a catastrophic collapse, seems to occur unpredictably in these models. This means that a single count of the number of ON words in a model does not tell us very much about how many active words we would expect to find in another count taken a few updates later. This discontinuity has some important implications for vocabulary testing in general, and for measures of attrition in particular.

The third important idea is that individual differences between conditions seem to be very large in these models. This idea is important because normal research does not often concern itself with single subjects, preferring to work with groups instead, pooling the data from many different participants. The models we have used in this workshop all have the same basic characteristics, so we might expect that they would show very little in the way of individual differences. You can explore this idea by running Program-13 again. Set the network parameters to nEv = 200 and rEv = 1234. Next run the model with these values for the NTWK parameter: 5001, 5004, 5014, 5040, 5041, 5050, 5053, 5066, 5069 and 5080. These parameter values will give you a set of ten simulations which all start with a high number of activated words. The program will implement 200 attrition events for each network.

You should find that all the cases show the catastrophic collapse that seems to be characteristic of the models in this simulation set, and this strongly suggests that vocabulary attrition is generally non-linear (see Schmid & Mehotcheva, 2012 for some relevant experimental data). However, Figure 11 shows what happens when we aggregate these simulations into a single data set. The figure shows the average number of activated words in the ten simulations after 25, 50, 100, 125, 150, 175 and

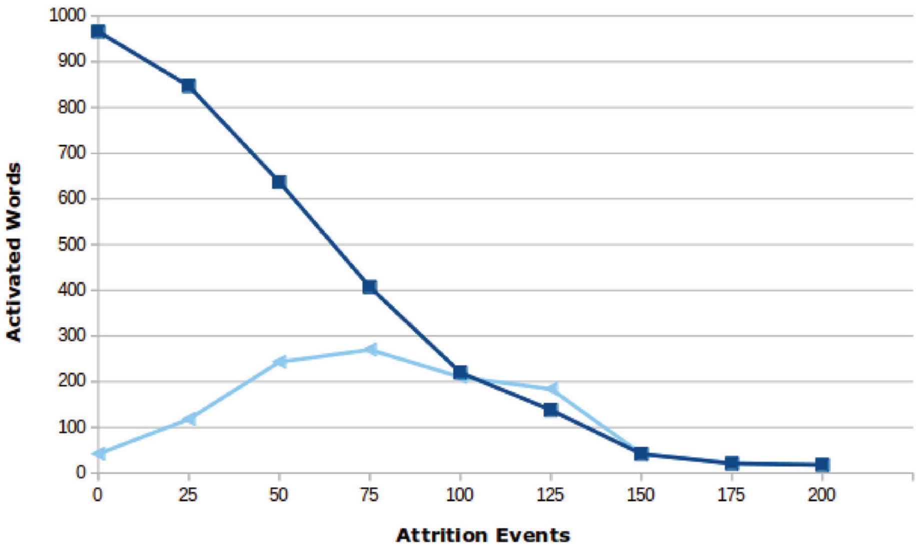

Figure 11 *The Mean Number of Activated Words in a Set of Ten Networks After a Series of Attrition Events are Implemented. (Program-13, with Parameter Settings nEv = 200, rEv = 1234).*

200 attrition events take place (dark blue line). The figure also shows the size of the standard deviations at each of these testing points (light blue line).

The obvious interpretation of Figure 11 is that attrition is gradual, and probably follows an exponential course. There are many examples in the literature where attrition is discussed in these terms (see, for example, Hansen, 1999). But it will be immediately obvious that the group data does not pick up the importance of catastrophic collapse in the simulations. None of the individual cases looks like the average data reported in Figure 11. Although the group scores are initially very homogeneous with a standard deviation of only 25 points, they become increasingly divergent as the simulations progress. After 75 attrition events, the average number of activated words has fallen from 966 to 407, but the range of the scores at this point in time is huge (highest score: 755 activated words, lowest score: 25 activated words; st.dev: 270 words). Only after 150 attrition events have taken place do the individual scores come together again (highest score 146 activated words; lowest score 3 activated words; st.dev: 22 words). At the very least, these data suggest that we need to be cautious about interpreting attrition data where only a single data collection point is reported.

The fourth important idea is that it matters how we measure vocabulary knowledge in attrition studies. In all these simulations, we have reported how many words in the vocabulary are ON after each update of the network. Intuitively, this feels like an obvious way to assess vocabulary knowledge – a common way of assessing vocabulary in older adults is simply to show them a small set of pictures and ask for the pictures to be named (e.g., Kaplan et al., 1983). However, the simulations developed in this Workshop suggest that this approach may not be a very reliable way of identifying the effects of attrition events, since most attrition events do not result in immediate vocabulary loss. A further problem is that real-life studies of lexical attrition are not able to test all the words a speaker knows. Instead, they rely on a small sample of the words in a vocabulary – and sometimes these samples are very small. We will explore this idea further in the next workshop.

A fifth idea to emerge from this workshop is that the way we have modelled attrition in Program-13 and Program-14 means that a new type of word has begun to appear in our network models. These zombie words – words that have very high activation thresholds – cannot be activated by actions that take place within the network, though they can be activated by an external event. Note that words of this type are not built into the network, but they emerge as a new word type as a result of other small changes to the network. We will explore this idea further in the next workshop, too.

Finally, the simulations in this workshop might be making you think about the time course of a set of attrition events. How long does it take to effectively deactivate a language that you once knew quite well? What sort of timescale is implied by the simulations, and how long would attrition events that we have modelled here take in real life? It is difficult to answer this question with any degree of confidence, but it is instructive to try to work out the arithmetic implied by the model. Let us suppose that the whole process of deactivating a vocabulary is typically complete in 6 years or 72 months. That would imply that our attrition events are taking place at a rate of roughly three or four per month, or about one every ten or eleven days. The implication of this is that attrition events may not occur very frequently in real life: rather, what is important is the cumulative effect these events have on the structure of the vocabulary in which they occur. This strongly reinforces the view that the simulated lexicons we have been studying in this paper are surprisingly vulnerable to infrequent small-scale changes, as long as these events take place repeatedly over a longish timescale. It also suggests that studies assessing attrition over relatively short timescales may be sensitive to fundamentally different attritional characteristics than long-term studies where attrition is studied over a period of months or years.

Conclusion

In this workshop we have looked at two ways of simulating attrition in a simple model vocabulary network. Once again, we have found that the vocabulary network idea seems to be much more important than we might have expected. Specifically, we have seen that attrition events do not always result in overt vocabulary loss. Rather, they weaken the overall structure of the network until it is no longer able to be a self-supporting structure. We also noted that a surprisingly small number of attrition events are needed before a catastrophic collapse in vocabulary activity becomes inevitable. These findings have some important consequences for real-life studies of vocabulary loss, and we will follow up these ideas in the next workshop.

References

Bahrick, H. (1984a). Semantic memory content in permastore: Fifty years of memory for Spanish learned in school. *Journal of Experimental Psychology: General* 113, 1–29. https://doi.org/10.1037/0096-3445.113.1.1

Bahrick, H. (1984b). Fifty years of second language attrition: Implications for programmatic research. *Modern Language Journal 68*(ii), 105–118. https://doi.org/10.2307/327136

Bahrick, H. P., Bahrick, L. E., Bahrick, A. S., & Bahrick, P. E. (1993). Maintenance of foreign language vocabulary and the spacing effect. *Psychological Science*, *4*(5), 316–321. https://doi-org.sire.ub.edu/10.1111/j.1467-9280.1993.tb00571.x

Bahrick, H., Hall, L. K., Goggin, J. P., Bahrick, L. E., & Berger, S. A. (1994). Fifty years of language maintenance and language dominance in bilingual Hispanic immigrants. *Journal of Experimental Psychology: General, 123*(3), 264–283. https://doi.org/10.1037/0096-3445.123.3.264

Hansen, L. (Ed.). (1999). *Second language attrition in Japanese contexts*. Oxford University Press.

Kaplan, E., Goodglass, H., & Weintraub, S. (1983). *The Boston Naming Test*. Leary Publications Company.

Kauffman, S. A. (1993). *The origins of order. Self-organisation in selection and evolution.* Oxford University Press.

Larson-Hall, J. (2019). L2 Lexical attrition. In M. S. Schmid & B. Köpke (Eds.), Schmid, M. S., & Köpke, B. (Eds.). *The Oxford handbook of language attrition* (pp. 377–390). Oxford University Press.

Nation, I. S. P., & Beglar, D. (2007). A vocabulary size test. *The Language Teacher, 31*(1), 9–13.

Schmid, M. S., & Mehotcheva, T. (2012). Foreign language attrition. *Dutch Journal of Applied Linguistics, 1*(1), 102–124. https://doi.org/10.1075/dujal.1.1.08sch

Weltens, B., K. de Bot, K., & van Els, T. (Eds.). (1986). *Language attrition in progress*. Foris.

Weltens, B., & Grendel, M. (1993). Attrition of vocabulary knowledge. In R. Schreuder & B. Weltens (Eds.). *The bilingual lexicon* (pp.135–156). John Benjamins. https://doi.org/10.1075/sibil.6.08wel