Vocabulary Learning and Instruction

Volume 11, Number 1,

December 2022
# Table of Contents

<table>
<thead>
<tr>
<th>Articles</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Letter from the Editors</td>
<td>iv</td>
</tr>
<tr>
<td><em>Joseph P. Vitta and Christopher Nicklin</em></td>
<td></td>
</tr>
<tr>
<td>A Proposed Taxonomy of Test-Taking Action and Item Format in Written Receptive Vocabulary Testing</td>
<td>1</td>
</tr>
<tr>
<td><em>Jeffrey Martin</em></td>
<td></td>
</tr>
<tr>
<td>Vocabulary Networks Workshop 1: Introduction</td>
<td>17</td>
</tr>
<tr>
<td><em>Paul Meara and Imma Miralpeix</em></td>
<td></td>
</tr>
<tr>
<td>Vocabulary Networks Workshop 2: Activating Words in a Network</td>
<td>35</td>
</tr>
<tr>
<td><em>Paul Meara and Imma Miralpeix</em></td>
<td></td>
</tr>
</tbody>
</table>
Dear VLI Readers,

Allow us to begin by wishing all of you a very happy New Year and the journal team wishes you a very successful and prosperous 2023. In this issue, we are pleased to present you with three original articles that are sure to stimulate both researchers and teachers with an interest in vocabulary. In the first article, Jeffrey Martin proposes a Taxonomy of Test-taking Actions Afforded by Receptive Vocabulary Test Format that acts as a heuristic to evaluate the influences of test format on written receptive vocabulary assessment. In the second and third articles, Paul Meara and Imma Miralpeix present two vocabulary workshops featuring a user-friendly technology that allows interested parties to model words in proposed networks. The first of these reports introduces the relevant concepts and technology while the second presents an example using 1000 words. These papers facilitate an introduction to connectionism principles within the arena of L2 instructed vocabulary.

We would like to conclude by thanking our editorial board and reviewers for their hard work this year and look forward to the new year.

With warm regards,

Joseph P. Vitta and Christopher Nicklin (Associate Editors) on behalf of the entire VLI team
A Proposed Taxonomy of Test-Taking Action and Item Format in Written Receptive Vocabulary Testing

Jeffrey Martin
Momoyama Gakuin University

Abstract
The functioning of a vocabulary testing instrument rests in part on the test-taking actions made possible for examinees by item format, an aspect of test development that warrants consideration in second-language vocabulary research. For example, although iterations of the written receptive vocabulary levels test (VLT) have integrated improvements in lexis sampling and distractor-item creation (i.e., Beglar & Hunt, 1999; Nation, 1983, 1990; Schmitt et al., 2001; Webb et al., 2017), its clustered form-meaning matching format has remained fundamentally unchanged. This study qualitatively explores the influence of this test item format on test-taking actions observed when taking the updated vocabulary levels test (UVLT, Webb et al., 2017). Data from a think-aloud protocol and retrospective interviewing indicated the predominant use of test-taking strategies for answering test items on the UVLT, such as bidirectional matching and elimination of cluster options, and that these actions enabled correct responses for clusters of target vocabulary about which the test taker demonstrated partial or even no knowledge. This evidence at the interface of test taker and test draws attention to the interconnection of estimating learners’ vocabulary knowledge and the action possibilities provided by item format on vocabulary tests. Such affordances are hierarchically structured in a proposed Taxonomy of Test-taking Actions Afforded by Receptive Vocabulary Test Format as a heuristic to evaluate the influences of test format on written receptive vocabulary assessment.

Keywords: receptive vocabulary knowledge, test modality, think-aloud protocol, affordance, heuristic

1 Introduction
Item format has become a salient aspect of second-language (L2) vocabulary testing due to a range of empirical research demonstrating how test format appears to tap different aspects of L2 vocabulary knowledge (VK) (e.g., Kremmel & Schmitt, 2016; Laufer & Goldstein, 2004; McLean et al., 2020). Schmitt et al. (2020) highlighted the need for rigorous vocabulary testing for specified purposes and that item format is one essential factor. The wide variety of vocabulary testing that is now commonly available includes the recalling and recognizing of lexis, the eliciting of lexical form and lexical meaning, and the selecting and matching of
options to solve for test items. Differences in testing outcomes bear implications about what can be inferred from test scores regarding the VK that L2 learners can employ within given settings of L2 communication.

To further explore this issue, direct observation of test-taking behavior can help to clarify what test-taking actions are enabled within an item format and to then make inferences about test-taking actions enabled by other item formats. This study centers on the observation of an assessment performed on a vocabulary test featuring an item-clustered form-meaning matching format. In this article, I begin by outlining the importance of item format on test scores. I also outline the importance of directly observing the actions taken on an item-clustered format vocabulary test, as well as a review of the procedure. I then present a set of actions observed to be made possible within this item format and the variety of these actions that were realized in answering different clusters. Finally, based on the analysis of these observed patterns, I propose a taxonomy of test-taking actions by test item format that hierarchically categorizes a range of target and non-target VK argued to be usable to solve for given item types.

2 Background

Vocabulary development is a multifaceted process for L2 learners (Nation, 2020; Read, 2020; Schmitt et al., 2020). Research suggests that VK elicited by vocabulary tests is partial and dependent on test item format (Kremmel & Schmitt, 2016; Laufer & Goldstein, 2004; McLean et al., 2020) and, therefore, may not be uniformly employable across distinct kinds of language performance (i.e., VK that facilitates reading a book versus writing a letter). To best meet the needs of pedagogic and research purposes, “careful thought needs to be given to the item type that is used to make sure that it is suited to the kind of knowledge it is supposed to measure” (Nation & Webb, 2011, p. 219). Thus, two important considerations for selecting or developing a vocabulary test are that the instrument can (a) reasonably target intended VK in test takers and (b) sufficiently estimate the presence of that knowledge.

Although starting from an exploratory stance, the following range of format types became relevant to this study. One common vocabulary modality of item formatting is recognition testing, where test takers recognize and choose a correct option for a test item listed among a set of distractor items. Another common test modality is recall testing, where test takers recall information to correctly answer test items without answer options to choose from. Additionally, these two test modalities are commonly formatted to elicit the specific aspects of receptive VK (e.g., L2 word meaning recognition) and productive VK (e.g., L2 word form recall). For instance, Schmitt (2014) presented evidence that producing L2 form is a “deeper” level of vocabulary mastery than the comprehending of L2 word meaning encountered in L2 input. An additional item format is a multiple matching item type that involves matching L2 word forms and L2 word meanings within a cluster of target vocabulary and definitions.

This study centered on the functioning of the item-clustered form-meaning matching format of the updated vocabulary levels test (UVLT; Webb et al., 2017).
The test was designed to be “a measure of receptive VK indicating the degree to which test takers may be able to understand the meanings of words that they encounter in written text” (p. 57). It was stated to not measure productive VK nor the influence of word frequency on word difficulty. The UVLT is the most recent iteration of the vocabulary levels test (VLT). Its 150 target words are sourced from the top five frequency bands of Nation’s (2012) British National Corpus (BNC)/Corpus of Contemporary American English (COCA) word family. Each band is represented by 10 clusters, and each cluster features three target words. The UVLT is preceded by Nation (1983, 1990), Beglar and Hunt (1999), and Schmitt et al. (2001). All are commonly structured in item clusters where test takers match six words (three target words and three distractors) with three definitions.

Quantitative studies have brought insight into the functioning of the VLT. For example, Kamimoto (2014) made an experimental version of the VLT by Nation (1990) that combined sets of three clusters to make larger clusters of 18 words and 9 definitions. This was in order to compare the relative effect of cluster size on testing outcomes. He found a nearly 19% inflation in test scores from the original test compared to the experimental version. In practice, estimating the lexical challenge an L2 learner might encounter with a given reading text would be hampered by a 19% inflation of assessment when considering the narrow thresholds for L2 reading referenced in the literature (e.g., 98% of words in a text known to facilitate reading fluency [Nation, 2006]). Additionally, Ha (2022) found correlations between correctly answered word items within clusters of the UVLT. However, an itemized language test should maintain test item independence in order to ensure fairness. This is to avoid disproportionately favoring test takers who answer select items correctly, over others who do not, by preferentially assisting them in correctly answering subsequent items (Bond et al., 2020). Score inflation and item interdependence interfere with the meaningfulness of test scores.

Directly observing clustered-format vocabulary testing is also considered due to the outcomes of Rasch-based analysis (Rasch, 1960), as was used by Beglar and Hunt (1999), Schmitt et al. (2001), and Webb et al. (2017). Webb et al.’s (2017) preliminary validity evidence for the UVLT found predictable item fit figures within the routinely accepted threshold of two standard deviations of the standardized mean (Bond et al., 2020). Notably for person measures, however, the threshold for removing test takers giving highly unpredictable/misfitting responses was inclusive of up to five standard deviations (Rasch outfit cutoff $z > 5.0$; Webb et al., 2017, pp. 38–39). For items and persons, the Rasch model anticipates that higher ability test takers will more likely respond correctly to items from high to low difficulty, whereas lower ability test takers will more likely only respond correctly to relatively easier items. If the opposite regularly occurs within the data, it may suggest that the test is garnering unintended or unpredictable test-taking behavior.

Comparing Rasch item reliability figures between vocabulary tests may not signal substantial differences in unintended test-taking behavior due to ceiling effects (upper limit of 1.0). However, Rasch item separation values scale with no upper limit and are not subject to ceiling effects (Smith, 2001). They represent statistically separable levels of difficulty instantiated by item functioning. Responses not accounted for by the model can degrade item separation figures, which lessens
confidence in the test’s replicability. Item separation is largely affected by the quality of the items, given a sufficient number of items comprise the test (Bond et al., 2020).

Webb et al. (2017) stated that the two forms of the UVLT (versions A and B) had reliability (and separation) estimates of 0.96 (4.72) and 0.96 (4.81), respectively. The UVLT is comparable to the vocabulary size test (VST; Beglar, 2010) if only the higher frequency target items are considered (e.g., a shortened version covering levels 1k – 4k). The VST is a written meaning-recognition test of 140 items that estimates the total size of a learner’s vocabulary: ten words each per fourteen 1000-word bands of the BNC word list by Nation (2006). Beglar (2010) demonstrated consistent item reliability and item separation for the full test (1k – 14k, 0.96 and 5.22) by comparing it to a subset of the test’s items. A version of only the first four frequency levels (1k – 4k, only 40 items) showed favorable item reliability and separation figures of 0.98 and 6.25. Compared to the UVLT (150 items, top five frequency levels), the separation figures indicate a 30% increase in the distinguishability of item difficulty for the shortened VST. The VST is a meaning-recognition test, and the UVLT is a clustered-matching test. The sampling of the VST and UVLT was comparable (BNC and BNC/COCA word lists). This evidence for differences in test functioning is limited but it invites a qualitative look into the actions enabled (afforded) by clustered vocabulary items.

Termed by Gibson (1979), affordances are what the environment “offers the animal, what it provides or furnishes, either for good or ill”, and evolutionarily, “they are unique for that animal” (italics in the original; pp. 119–120). Gibson’s theory of affordance in nature was applied by Norman (1988) to user interface design in human–computer interaction, where actionability with objects or computer screens becomes salient depending on product design and user perception. Norman’s (1988) conceptualization of affordances includes the influence of cultural conventions in how humans perceive objects, which are subject to the physical constraints of the objective existence of affordances in the given environment. This conceptualization is taken to be applicable to actions and strategies afforded by item format as a test taker interfaces with a paper-based vocabulary test.

Affordance is also conceptualized within the sociocognitive approach to SLA (van Lier, 2004). For example, Atkinson et al. (2018) studied the activity of collaborative baking, “a form of triadic interaction, wherein individuals focus shared attention and action on co-active environmental affordances in completing a task” (italics in the original; p. 477). Churchill et al. (2010) conducted a sociocognitive study of the interaction between a tutor and a student working together on a grammar worksheet. These studies investigated affordances within social interaction. In contrast, the current study is about one participant’s engagement with a vocabulary test, so affordance is bound to Norman’s conceptualization (1988).

The primary source of data for analysis in this study was a think-aloud protocol (Johnstone et al., 2006), where test takers verbalize their thought processes as they complete a task. Wilson’s (1994) paper on the completeness of data retrievable by a think-aloud protocol highlighted the inability to elicit unconscious thoughts, nor all conscious thoughts, of participants, but that its concurrent verbal
reporting held advantages of accuracy and richness over retrospective questioning. Topic is also important because during a think-aloud protocol, “self-presentation concerns are more likely to be operative in […] social domains” than in problem-solving domains due to social topics being potentially more sensitive (p. 251). The problem-solving task of completing a vocabulary test in this study seems appropriate for a think-aloud protocol. As a complimentary data source, interview data can also attest to a test taker’s thoughts and actions (Schmitt, 1999; Schmitt et al., 2001).

An exploratory investigation of test-taker actions and testing outcomes holds no preconceptions. The notion of affordance in this study became relevant during the analysis stage as a way to categorize the actions that were observed to be made possible by the clustered item format. A resulting taxonomy represents a hierarchical structure of test-taking actions in relation to the written receptive vocabulary testing formats summarized above. The proposed taxonomy details relationships of affordances, and it may provide a heuristic that is generalizable within vocabulary testing research. It is open to be rebutted or corroborated in proportion to the weight of the evidence presented. The considerations detailed above drove the formulation of the research questions given as follows:

**RQ1:** What lexical knowledge does a test taker demonstrate on the UVLT?

**RQ2:** What test-taking actions are afforded to a test taker when taking the UVLT?

**RQ3:** How and to what extent can the data about vocabulary knowledge and test-taking actions lead to categorizations and inferences regarding the formatting of written receptive vocabulary testing?

### 3 Methodology

#### 3.1 Participant

The participant was a 27-year-old Japanese female working at the patient service counter at a hospital in the Tokyo area. Naoko, a pseudonym, had never taken the UVLT or any prior VLTs. She reported earning a TOEIC score of 420 points during her second year at university. She also reported studying English since that time for social and travel reasons. Naoko and I met to discuss the study, which would entail three meetings over a 2-week period, and the compensation would include a gift card. She gave her informed consent to participate in the think-aloud protocol, to be interviewed, and to be audio recorded. All data and field notes were kept in a secure location offline.

#### 3.2 Materials

The UVLT (Webb et al., 2017) is a 150-item test sampled from the five most frequent 1000-word family bands of the BNC/Coca family word list by Nation (2012). The two versions (A and B) of the UVLT were developed to be
It should be answered in the following way.

<table>
<thead>
<tr>
<th>land with water all around it</th>
<th>game</th>
<th>island</th>
<th>mouth</th>
<th>movie</th>
<th>song</th>
<th>yard</th>
</tr>
</thead>
<tbody>
<tr>
<td>part of your body used for eating and talking</td>
<td>✔</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>piece of music</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
</tr>
</tbody>
</table>

Figure 1. Example of the cluster of items on the UVLT (Webb et al., 2017).

Figure 2. Example of the cluster of items on the VLT (Schmitt et al., 2001).

equivalent. The UVLT Version B was used in this study and is henceforth referred to as the UVLT. Each 1000-word band is represented by a 30-word sample in ten clusters, each including three target words. Each band consists of five clusters of nouns, three clusters of verbs, and two clusters of adjectives. Each cluster contains three definitions in English and six vocabulary items (three target words and three distractors). An instructional example from the UVLT is shown in Figure 1. An example of the VLT (Schmitt et al., 2001) is also provided in Figure 2 to illustrate the continuity of the clustered matching format across iterations of the VLT. The clusters were numbered in this study to help illustrate findings, representing the \( n \)th cluster as ordered on the UVLT version B. The test is accessible at https://www.edu.uwo.ca/faculty-profiles/docs/other/webb/NVLT-VERSION-B.pdf.

### 3.3 Procedure

The sequence of Naoko’s participation in the study is detailed in Table 1. First, Naoko completed the odd-numbered clusters, 25 of the 50 clusters, at her own pace. She was informed that the interview may cover her answers of the odd-numbered items. One week after completing the odd-numbered clusters, Naoko completed the even-numbered clusters while following a think-aloud protocol (Johnstone et al., 2006; Wilson, 1994), where she was asked to concurrently verbalize her thought processes as she completed the clusters of items. In cases where Naoko stopped talking, I provided follow-up prompts such as, “What are you thinking now?” and “Please say what you’re thinking.” Naoko’s description of her decision-making process was transcribed. Her responses for all 150 items were tallied and reviewed. I then used these data to prepare for the retrospective interview.

The semi-structured interview was about how she engaged with the UVLT (revisiting clusters from both even- and odd-numbered clusters). She was free to participate in either English or Japanese. Examples of the interview questions for Naoko were “Please tell me what the word means?” and “How did you decide on that answer?”. Next, I transcribed the interview session and organized all data.
into spreadsheets for detailed coding and analysis. As a researcher interested in L2 vocabulary, I was aware of L2 vocabulary studies on partial word knowledge, the influence of cognates, and so on, but I held no preconceptions regarding Naoko’s engagement with the UVLT, nor any preconceived goals to later organize observed actions by item format. The resulting coding system was an emergent outcome of analyzing the study’s data. Coded transcripts for the think-aloud protocol and the interview are placed in online supplemental materials that are retrievable from the Open Science Framework (OSF; https://osf.io/ypxze/).

3.4 Analysis

Analysis of the transcribed data saw the emergence of patterns for coding using frameworks outlined by Saldaña (2016). Initially, the data were coded for attributes such as cluster number and frequency band. Next, coding for the evidence of VK and patterns of action emerged, and this led to the creation of a code book (presented in supplemental materials). The actions observed in Naoko’s test taking suggested that she was making decisions in a hierarchy of patterns, with some leveled above others. Such data called for taxonomic coding, which Saldaña (2016) described as a way to discover the “knowledge that people use to organize their behaviors and interpret their experiences” (p. 157). The resulting taxonomic coding of the think-aloud data and interview data was revised over repeated analysis.

After coding the data, the data and codes were shared in a data session with seven colleagues who were familiar with the project and had experience in second-language studies. All together and in smaller groups, the members of this session analyzed the transcripts of the think-aloud protocol and the interviews. Their feedback confirmed much of the coding scheme and provided additional nuance to the coding and analysis.

4 Findings

4.1 Vocabulary Knowledge Demonstrated

Observed degrees of VK ranged from no VK to robust VK and observations varied by cluster. Partial word knowledge was evident in at least two ways: knowledge of word parts and knowledge from L1 loanword equivalents.
For example, the think-aloud data (lines 82–83) for cluster c38 showed that knowledge of the prefix “trans,” meaning “to change,” allowed Naoko to match the unknown word “transplant” to its meaning of “move something to another place.” Another example, loanword knowledge, enabled Naoko to eliminate the distractor word ‘tank’ in cluster c14. In the interview, I asked if she knew that word, and she responded, “Like water tank?”, in reference to one of its loanword equivalences in Japanese (Interview, lines 20–24). She did not describe additional meanings for tank, but her knowledge was sufficient to eliminate this distractor option.

Naoko spoke of having no VK for some target items during the think-aloud protocol session and the interview. For some of these items, she guessed incorrectly. For example, “Twenty-eight. Just my guess but ‘exceed’. ‘Goes beyond the limit’ is ‘decline’. ‘Take in’ is ‘link’. I don’t know these words. I don’t know the meaning” (Think-aloud, line 50). Nevertheless, numerous other items were answered correctly despite Naoko’s lack of VK for the target words. A prime example from the interview data was cluster c29, with the targets “approximate,” “frequent,” and “prior,” where Naoko stated having no knowledge despite successfully matching all three of the cluster’s form-meaning pairs:

The words are kinda like… yeah, I don’t really know them, but I knew some of the left side (definitions). Like maybe ‘happening often’ is frequency, but I wasn’t sure, ya know? But I didn’t know ‘approximate’ and ‘prior’… I didn’t know the meanings at all. I know ‘graphic and ‘vital’, but their meanings don’t fit with the left side. (Interview, lines 56–60)

For the remaining distractor word “pale,” she commented that she knew the color. With minimal or no knowledge of the three target words in cluster c29, Naoko correctly matched them to their meanings merely by eliminating cluster options.

Analysis of these data made it apparent that partial VK aided Naoko in matching form and meaning. For cluster c26, Naoko states, “Agree. consent, enforce, exhibit, retain, specify, target. (2-second pause) Hmm. I don’t know. Next, ‘say clearly’… (2-second pause) I don’t know anything” (think-aloud, line 44). From this segment of think-aloud data, it was not explicit whether she merely eliminated options, used unstated knowledge of the target words, or simply guessed, but she matched all items correctly for the cluster despite her stating that she had no knowledge. Naoko’s solving for this cluster resembles patterns of her solving for other clusters and generally illustrates the success she could achieve by using degrees of VK.

### 4.2 Actions Afforded to a Test Taker When Taking the UVLT

The format of the UVLT appears to afford test takers the additional ability to switch matching direction between linking target forms to meaning and the list of meanings to target forms. Naoko was observed to overwhelmingly process meanings listed in the left column of each cluster first and then to match them to target word forms in the top row of the cluster. Having a definition in mind and then selecting a target form from a list of options appears to tap a test-taker’s ability to recognize L2 word form, an aspect of word knowledge that is distinct from meaning recognition.
knowledge, as demonstrated by Laufer and Goldstein (2004) and Schmitt (2014). Naoko began cluster c26 with the meaning of “agree,” the first meaning listed in the cluster’s left column. Another example is cluster c2, where Naoko says “Body part that sees. It’s only ‘eye’. Parent who is a man. Parents are only fathers or mothers, so… part of the day with no sun… it means cloudy or just night… It’s night” (Think-aloud, line 2). Each of these instances started with the term on the left read first and then matched to the correct word form listed at the top.

In total, 84% of the even-numbered clusters (21 of 25) saw at least partial use of this strategy of reversed matching (Table 2). Switching that was not verbalized or noticed could have occurred for the four other clusters as well, but the full extent of switching is not observable. It is also possible to repeatedly switch. Bidirectional matching is at odds with the measuring of written receptive VK employable for reading because definitions are not provided first with words presented second in the act of receiving written texts. Although the aspects of vocabulary are interrelated, the accuracy of estimating pertinent VK is reduced when vocabulary measures and language performance measures are not ecologically aligned (McLean et al., 2020; Schmitt, 2014).

Another affordance that Naoko acted upon was eliminating options in order to solve for items. Table 3 details the observed behavior of option elimination and guesswork on clusters listed by the number of correct answers. The observed data suggested that she used elimination to varying degrees by cluster. Also, since each cluster was unique, Naoko’s partial knowledge specific to each cluster could have inflated her number of correct answers in idiosyncratic ways. From the think-aloud and the interview, it appeared that 72% of the even-numbered clusters (18 of 25) were answered using elimination. The incorrect answers for clusters c28 and c50 appeared to be total guesses. Clusters c2, c4, c6, and c8 appeared to be correctly answered without eliminating options. The clusters with no evidence of option elimination appeared to be clusters of words and definitions that were very easy or very difficult to understand for Naoko. Nonetheless, unobserved or unverbalized option elimination could have also been present, which would increase the weight of evidence.

As Wilson (1994) outlined, not all conscious thought is expressed in a think-aloud protocol. Additionally, not all recollections are expressed during an interview. Therefore, the actions of switching and elimination could have occurred to

---

Table 2. Observations of directions in target matching during think-aloud protocol

<table>
<thead>
<tr>
<th>Observed matching direction</th>
<th>Cluster and frequency band</th>
</tr>
</thead>
<tbody>
<tr>
<td>From target form to target meaning</td>
<td>c28 (3k) c32 (4k) c42 (5k)</td>
</tr>
<tr>
<td>c2 (1k) c12 (2k) c22 (3k) c34 (4k) c44 (5k)</td>
<td></td>
</tr>
<tr>
<td>c4 (1k) c14 (2k) c24 (3k) c36 (4k) c46 (5k)</td>
<td></td>
</tr>
<tr>
<td>From target meaning to target form</td>
<td>c2 (1k) c4 (1k) c6 (1k) c8 (1k) c10 (1k)</td>
</tr>
<tr>
<td>c2 (1k) c12 (2k) c22 (3k) c34 (4k) c44 (5k)</td>
<td></td>
</tr>
<tr>
<td>c4 (1k) c14 (2k) c24 (3k) c36 (4k) c46 (5k)</td>
<td></td>
</tr>
<tr>
<td>c6 (1k) c16 (2k) c26 (3k) c38 (4k) c48 (5k)</td>
<td></td>
</tr>
<tr>
<td>c8 (1k) c18 (2k) c30 (3k) c50 (5k)</td>
<td></td>
</tr>
<tr>
<td>c10 (1k) c20 (2k)</td>
<td></td>
</tr>
</tbody>
</table>

Note. Cluster number shown as “c4 (1k),” meaning Cluster No. 4 of the first 1000-word band.
an even greater extent, but even so, such unobserved action does not harm the evidence that was collected. Naoko was observed to switch matching direction between a mode of processing meaning from form and form from meaning. She was also observed to eliminate options to solve for clusters. Because Naoko did not seem to eliminate options to solve for the very easiest or the very hardest clusters, it may be that she was prompted to act at a threshold of difficulty but had to act within the limits of her knowledge of the given lexis. In taking the theoretical position of Gibson (1979) and Norman (1988), the bounds of this evidence invite the possibility that, when a test taker takes up the actions of switching matching direction and answering option elimination, it is the human response to this testing environment.

5 Discussion

5.1 Building a Taxonomy for Testing Receptive Vocabulary Knowledge

Naoko’s answers for target words on the UVLT suggested a hierarchy of action. It seemed warranted to organize these actions within a parsimonious structure. Actions repeated here from the code book were differentiated as follows:

**Strategies afforded by test format**

a. Guess an answer for unknown reason
b. Guess an answer from word part
c. Guess an answer from loanword knowledge
d. Select an answer by eliminating other options
e. Switching between matching word target to answer option and vice versa

A taxonomic structure emerged from the categorization of observed actions and VK. Patterns began to fit the affordances inferred to exist for the other commonly used written receptive vocabulary testing formats summarized above.

Saldaña (2016) detailed how a taxonomy can represent the actions of people in a setting. Spradley (1980) illustrated different forms that taxonomies take in social settings, such as tree diagrams, but he also described the actions of the lone person or even the functionality of an object by using taxonomic diagrams of various configurations (see pp. 114 & 120). In this tradition, the observed patterns were delineated as an expanding order of test-taking actions made possible by format. A taxonomy emerged that structured the affordances provided by the clustered form-meaning matching format. The meaning-recognition format was reasoned to not afford as many test-taking actions as the item-clustered format. The meaning-recall format was included but was reasoned to be most restrictive of test-taking actions.

In the Taxonomy of Test-taking Actions Afforded by Receptive Vocabulary Test Format (see Figure 3), affordances provided by item format are illustrated by three shaded boxes, each expanding outward from smallest to largest in size.
The various test strategies detailed in this study are all bounded within the largest box representing the clustered form-meaning matching format of the UVLT. This item-clustered format introduces a decision point to the test taker: the ability to switch between matching target form to target meaning and target meaning to target form. Naoko took up this affordance of switching matching direction, as shown in Table 3.

When matching target form to target meaning, the path in the taxonomy enters the middle box representing the meaning-recognition test format. This format allows the test taker to decide whether to eliminate options among a list of definitions using VK, even if partial or incomplete. The action of eliminating options is also afforded to test takers who switch their matching direction, but such action remains in the outer box because it would not belong within the middle box representing meaning-recognition testing (e.g., Schmitt et al., 2020).

The decision points are connected by arrows on the taxonomy. Switching could be repeated between the matching directions afforded by item-clustered test format. Switching could happen and return back unnoticed if not verbalized. For the meaning-recognition test format, the arrows within the box illustrate the logical result that once an option is eliminated, this action is not reversible: one could second-guess themselves, but a state of “yes” remains for that cluster. If the test taker does not act on the affordance of eliminating options, a test taker directly selects an option.

However, the presence of option choice keeps test-taking action outside the innermost box (meaning-recall test format). As a result, the taxonomy has no arrows entering the innermost box, which excludes the possibility of test takers to ignore the affordances introduced by the meaning-recognition test format and the

---

**Table 3. Option elimination and correctly answered items per cluster during think-aloud protocol**

<table>
<thead>
<tr>
<th>Observed behavior</th>
<th>Correct target items per 3-word cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3 (all) correct</td>
</tr>
<tr>
<td>Complete guesswork</td>
<td>c26 (3k)</td>
</tr>
<tr>
<td>No elimination observed</td>
<td>c12 (2k)</td>
</tr>
<tr>
<td>Stated to not know any words</td>
<td>c14 (2k)</td>
</tr>
<tr>
<td>(Eliminating at least one option)</td>
<td>c32 (4k)</td>
</tr>
<tr>
<td>Guess between options (&gt;2)</td>
<td>c38 (4k)</td>
</tr>
<tr>
<td></td>
<td>c40 (4k)</td>
</tr>
<tr>
<td></td>
<td>c10 (1k)</td>
</tr>
<tr>
<td>(Eliminating all but one option)</td>
<td>c18 (2k)</td>
</tr>
<tr>
<td>Select the remaining option</td>
<td>c18 (2k)</td>
</tr>
<tr>
<td>Directly select option</td>
<td>c2 (1k)</td>
</tr>
<tr>
<td>No elimination observed</td>
<td>c4 (1k)</td>
</tr>
<tr>
<td></td>
<td>c8 (1k)</td>
</tr>
</tbody>
</table>

**Note.** Cluster number shown as “c4 (1k),” meaning Cluster No. 4 of the first 1000-word band.
clustered form-meaning matching format. The meaning-recall test format only allows the test taker to enter L1 meaning for an L2 word form directly into a blank answer space. In other words, affordances provided by the formats of the outer two boxes preclude the ability to demonstrate written receptive VK at the level of meaning recall.

5.2 Heuristic for Evaluating Item Format

Emerging from this study’s findings, the *Taxonomy of Test-taking Actions Afforded by Receptive Vocabulary Test Format* in Figure 3 illustrates how researchers can narrow their L2 vocabulary measurement to knowledge that they theorize to be relevant to their studies. A review of the test formats entered into the taxonomy is warranted. Functioning within the middle box of the taxonomy is the meaning-recognition item format of multiple-choice testing. Although its purpose is not the same as the VLT, the VST by Beglar (2010), detailed in the literature review, is an example of a meaning recognition test (Figure 4). This format does not afford matching or bidirectional matching. The innermost box in Figure 3 represents meaning-recall testing (Figure 5), which does not afford option selection, nor option elimination, nor option matching (e.g., vocableveltest.org [McLean & Raine, 2019]).

Naoko’s test taking exhibited behaviors centered on eliminating options and the switching of form-meaning matching direction. Neither of these test-taking strategies aid in the processing of L2 input when actually reading. Given the affordance, test takers utilize partial knowledge to solve for items. The UVLT and its predecessors are often used as a measure in experimental research designs. As discovered in this study, test results can be realized in ways unknown and unintended to testing researchers’ purposes. Nevertheless, an accurate measure of
VK is important for experimental research and pedagogic aims with L2 learners. For example, VLT has been used to match L2 learners with lexically appropriate reading materials, and in the case of fluency building, 98–100% of the words should be known to ensure that lexical difficulty does not impede the processing of L2 input (Hu & Nation, 2000). Vocabulary test scores allow for estimates of lexical coverage in this case, which include an inherent margin of error, but a test maker should develop or select a test to reduce this error as much as possible.

Precision in estimating VK is ideal to avoid selecting L2 materials that are too easy or too difficult for a research or pedagogic purpose. Within a framework such as the four strands (Nation, 2007), there are additional pedagogic aims for known vocabulary in texts. Erroneous estimations of lexical coverage for study materials could find texts intended for fluency building to be too difficult and actually be suited for form-focused instruction (i.e., lexical coverage at 95%). The difference between 98% and 95% of lexical coverage equates to a difference of one or two extra words per 50 words not being known by an L2 learner. This narrow margin for error would be sensitive to an inappropriate testing format. The taxonomy outlines the influence of affordances that item format introduces. On the basis of vocabulary studies concerning item format that have been cited in this study and the observation data of this study, the proposed taxonomy can serve as a reference for researchers to further explore relevant issues of vocabulary assessment and development.

6 Conclusion

This study is rooted in data about the functioning of the UVLT gathered from a think-aloud protocol and retrospective interviewing. Qualitative and test score data were coded and analyzed. Patterns of affordances emerged, which led to a taxonomy of test-taking actions argued to be made possible to test takers by test item format. The expanding boxes of the taxonomy represent additional affordances that increase test-takers’ abilities to make use of degrees of VK and guesswork. In fact, the clustered form-meaning matching format (outermost box)
afforded instances of correctly answering for target items within clusters despite having no apparent knowledge of the target words. At three levels in the taxonomy (item-clustered format, meaning-recognition format, and meaning-recall format), VK inferred to be usable at each level is detailed.

The presence of affordances and use of partial VK narrows at each level in the taxonomy in a graphic form. It illustrates a hierarchy of testing conditions used in empirical studies about the effects of item format on measuring VK (Kremmel & Schmitt, 2016; Laufer & Goldstein, 2004; McLean et al., 2020). Furthermore, the hierarchy of test scores in these studies amounts to a convergence of evidence that format matters for written receptive vocabulary testing. When a test taker performs the actions of switching, matching, and eliminating, these are human responses to the testing environment. In sum, the taxonomy proposed in this exploratory study provides a structure from which item format can be viewed and evaluated given pedagogic or research aims of interest.

Acknowledgments

I would like to thank Dr. Tamara Swenson and other colleagues, as well as the two anonymous reviewers, for their constructive comments and suggestions, all of which helped improve the manuscript.

References


Kamimoto, T. (2014). Local item dependence on the vocabulary levels test revisited. Vocabulary Learning and Instruction, 3(2), 56–68. https://doi.org/10.7820/vli.v03.2.kamimoto


Vocabulary Networks Workshop 1: Introduction

Paul Meara\textsuperscript{a} and Imma Miralpeix\textsuperscript{b}

\textsuperscript{a}Swansea University; \textsuperscript{b}Universitat de Barcelona

Abstract

The idea that a vocabulary is a network of words is one that has become a common theme of the second language (L2) vocabulary research literature. However, not many people have considered the wider implications of this powerful metaphor. This paper is the first in a series of workshops that examines some of these implications. In this first workshop, we introduce some very simple computational models which illustrate some basic properties of network structures.

The workshop consists of a set of interactive programs that model a vocabulary as a Cellular Automaton—a minimally organized network where each word is linked to a small number of other words, and words change their activity status depending on the inputs they receive from other words. These networks exhibit some surprising emergent properties which have important implications for the way we understand and think about real vocabulary networks.

Keywords: Cellular automata, Emergent properties, Network structures, Simulations, Vocabulary modeling

1 Introduction

The idea that a vocabulary consists of a set of words that are linked together in a network is one that has become pervasive in the second language (L2) vocabulary research literature. However, not many people have looked at the implications of a networked vocabulary, or considered how a network might be different from a mere heap of words that are not connected to each other. Despite Aitchison’s assertion that “words cannot be heaped up randomly in mind” (Aitchison, 2012:5), most discussions of vocabulary networks in the L2 research literature are not much more than a vague metaphor. Occasionally, we find a reference to Collins and Quillian’s idea that a vocabulary network might work on “spreading activation”: when a word is activated it also activates other words that it is connected to, and this causes a ripple of activation to spread through the entire lexicon (Collins & Loftus, 1975). This is an attractive idea, but, in the L2 literature at least, the details of the mechanism are rarely specified, and this makes this research difficult to evaluate.

Most L2 vocabulary research assumes that the “network” which underpins a vocabulary is basically the same as the networks that we find in word association studies. Again, this looks like a reasonable assumption at first sight, but word association studies tend to use relatively small stimulus sets, and it is not easy to see how a network built out of the responses to, say, 50 stimulus words might scale up to a much larger vocabulary of several thousand words.
This workshop is an attempt to address these issues. The approach taken in the workshop is a rather unusual one, in that it does not involve collecting empirical data from real language learners. Instead, it uses simulations—computer models of vocabulary-like networks which look as though they might provide useful analogues for real vocabularies. This approach turns out to be a very useful tool for L2 vocabulary researchers (e.g., Segalowitz & Bernstein, 1997): its main advantage is that it removes some of the messiness that we usually find in ordinary empirical studies, and allows us to examine in detail some of the assumptions that most of us take for granted. Pulling apart ideas that we usually do not question is almost always a good thing for researchers to do, as it helps to rethink ideas about L2 vocabulary acquisition. In addition, as we shall see, working with simulations is an extraordinarily effective way of generating interesting speculations about how L2 vocabulary networks function in real life.

2 How the Workshop is Structured

The workshop is organized around a series of questions of increasing complexity. We will be examining the questions by working with some specially prepared computer programs that are available on the workshop website.

The workshop is mainly interested in L2 vocabulary networks, but we will start off with a basic introduction to building vocabulary models and running simulations. To start with, we introduce you to a minimal type of network that will serve as a preliminary model of how a vocabulary network might work. Networks of this type are deceptively simple, but they turn out to have some interesting emergent properties that may have important implications for our understanding of real L2 vocabulary networks. Then, we will progressively work with more complex models with several parameters that you can change, allowing you to interact with the network in many different ways. After that, we will focus on specific aspects that can be studied using model networks: vocabulary attrition and loss, bilingual and multilingual lexicons and how they interact with each other, and how we might grow a vocabulary from scratch.

Simulation research has had something of a bad press in the Applied Linguistics research literature, on the grounds that it is so far removed from “the real world” that it is just irrelevant. This workshop is designed to show just how short-sighted this view is. We hope that running these simulations will teach you a very different way of asking questions about vocabulary, and that your own research will be enriched as a result.

3 Part One: First Simulations

3.1 Basic Networks

This section will provide some essential background to a set of simple network models called Cellular Automata. These models usually consist of a set of objects that interact with each other in some way. Modeling a vocabulary in this way looks as though it should be fairly straightforward. Each word in the
Vocabulary is an object that is linked to a number of other words, and all we need to do in order to set up a basic model is to be specific about the connections and how they allow interactions between the words. We will start off by looking at some very small model vocabularies so that you can familiarize yourself with the type questions we will be asking in later parts of the workshop, when we look in more detail at larger, more realistic vocabularies.

We can illustrate the basic approach with the simple network shown in Figure 1.

In this network, we have five words. Each of the words can be in one of two activation states, conventionally called ON and OFF. In addition, each word is connected to one other word. These connections (shown by the arrows), are fairly limited in what they can do: the connection from WordA for instance simply sends a message from WordA to WordB that tells WordB whether WordA is ON or OFF (we call this message “input”), and the connection from WordB just sends a message to WordC that tells WordC whether WordB is ON or OFF. And so on. WordE does not have any outgoing connections.

In Figure 1 all the words start out in the OFF state, so the network does not actually do anything. But we can make this network more interesting by specifying how each word responds to its input. Let us say that if a word receives a message telling it that its input is ON, then it too will go ON. In Figure 1, if WordA is briefly turned ON, then WordB “knows” this, and will turn itself ON, sending a signal to WordC as it does so. Basically, in this network, if a word receives an input from its neighbor, it becomes activated and goes ON. If a word is not receiving an input, it goes OFF.

Now consider what this network does if we briefly activate WordA: WordA will go ON, and because WordA is ON, it sends a signal to WordB. This has the effect of turning WordB ON. Meanwhile, WordA is not receiving an input anymore, so it goes OFF. We now have one activated Word, WordB, and this word sends a signal to WordC, so WordC turns itself ON. WordB meanwhile turns itself OFF. WordC sends a signal to WordD, and turns itself OFF. This signal turns WordD ON, and sends a signal to WordE. WordE turns itself ON briefly, but does not do anything else. WordD will have turned itself OFF, so WordE is no longer getting a signal and will turn itself OFF. In this way, activating WordA will send a pulse of activity through the network. Each word will be activated in turn, and when the last word has been activated, the pulse will stop, and the network will return to its resting state, where no words are ON.

It is fairly straightforward to work out the behavior of this network just by looking at it, and examining how each word impacts on all the other words. With a small network, you can usually do this in your head—or at least with a pencil and
paper. However, if we add more connections between the words, then the behavior of the network becomes slightly more complicated. You can see this in Figure 2.

Here we have added just one new connection between WordE and WordA. But this new connection fundamentally changes the way the network behaves. If we turn WordA ON, then the pulse of activation runs through the network as usual, but when WordE gets turned ON, it sends a signal back to WordA and the whole process starts all over again. In this model, the network finds itself in a never-ending loop, where a pulse of activation goes round and round the network forever. Moreover, WordA does not have a special status in this model: turning any of the words ON will start the pulse of activation.

Now look at Figure 3. Before you read on, try and work out how this network behaves if WordA is activated.

In Network 3, activating WordA sends a signal along the network, activating each word in turn. When the signal reaches WordE two outputs are sent out: one goes to WordA, the other goes to WordC. When the network updates itself, both these words are then activated, so we now have two signals in the network, rather than one. The next time the network updates itself, both these signals will propagate through the network. WordA being ON will turn WordB ON, and WordC being ON will turn WordD ON. This will move the network into a new state of activity where WordA is OFF, WordB is ON, WordC is OFF, WordD is ON and WordE is OFF. The next time the network updates itself, it will move into a new configuration where WordA is OFF, WordB is OFF, WordC is ON, WordD is OFF and WordE is ON.

Now things start to get much more complicated. You should be able to work out that the network will now move to a configuration where it has three activated Words: WordA will go ON (because it is linked with WordE), and WordD will go ON (because it is linked to WordC). WordC would normally go OFF, but at this...
time it is receiving an input from WordE, so it will remain ON. These changes will leave the network in a configuration where WordA is ON, WordB is OFF, WordC is ON, WordD is ON and WordE is OFF. Can you work out what happens next time the network is updated?

You probably found this description confusing and difficult to follow. To be honest, it is really hard to work out in your head how even a very small network behaves. Therefore, it is not surprising that various methods have been devised to show what the network is doing. We have illustrated two of these methods in Figures 4 and 5.

Figure 4 summarizes the behavior of Network 3 when a single word is activated. In this figure, the five words in the network are shown as a horizontal line of 1s and 0s. Each horizontal line summarizes one step in the model. When a word goes ON, it is shown as a red 1; when a word is OFF, it appears as a blue 0. At time 0, there is no activity in the network: all the words are OFF. At time 1, WordA has been turned ON. The remaining lines show how the network responds to this event. This illustration shows very clearly how the network goes from a very low level of activity to a state where all its nodes are ON. Once the network reaches this final state, it stays there forever.

<table>
<thead>
<tr>
<th>Time</th>
<th>WdA</th>
<th>WdB</th>
<th>WdC</th>
<th>WdD</th>
<th>WdE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 4. How Network 3 Changes Over Time.
Figure 5 shows a different way of thinking about the behavior of a network. In this figure, we have a trajectory map where the individual cells in Figure 4 have been collapsed into a single row: a string of 1s and 0s. Therefore, the state of the network at time1, with just WordA active, can be described as [10000]. At time2 the network moves into a new state where only WordB is active, and this can be described as [01000]. At time3, the network is in state [00100]. And so on. Each state the network finds itself in has a unique successor state. So, if the network in Figure 3 starts off in state [00001], then it will always move itself to state [10100].

When you have five words that can take on one of the two values, there are $2^5 = 32$ possible combinations of 0s and 1s. Figure 5 shows each of these 32 states, each linked to its unique successor state, in the form of a trajectory map. It is very easy to see from Figure 5 that there are only two possible long-term outcomes for Network 3. If the network starts out in state [00000] then it will do nothing. All the nodes will remain inactive. On the other hand, if it starts out in any other state, then the network will inevitably end up in state [11111], where all the words are activated. Sometimes this will happen quickly—for instance, if we put the network into state [10111] then it will immediately move to state [11111] and...
stay there. On the other hand, if we start the network off in state \([10000]\) (at the bottom of the Figure) then it will eventually reach state \([11111]\) after moving through the 14 intermediate steps shown in Figure 4.

State \([00000]\) and state \([11111]\) are examples of attractor states—states where the network is in a stable configuration, and does not move to a new state. As we will see in the later sections of this workshop, attractor states turn out to be an important concept when we examine the way network models of vocabulary behave.

### 3.2 Vocabulary Networks

You might now be wondering why we have spent so much time on a tiny model which clearly has nothing to do with L2 vocabularies. The answer is that even this tiny model illustrates some important differences between a heap of words and a network of words. Basically, a heap is inert, whereas a network is usually dynamic, and the behavior of networks can be surprisingly complex. In Network 3 we have only five words and only six links connecting them together, but already it should be obvious that even a network as small as this embodies a huge amount of complexity. And if you tried to work out for yourself how Network 3 responds when it is activated, then you will have realized that it is almost impossible to do the necessary calculations in your head. You can just about manage it when you have only five words in the network, but anything bigger than this rapidly becomes unmanageable.

Real vocabulary networks are obviously a lot bigger than Network 3, and their behavior will be correspondingly more complex. You can get a feel for this by doing some thought experiments in your head. For example, think what would happen to Network 3 if it was in state \([11111]\) and one word gets deactivated. Try to work out how Network 3 would behave if you added another connection to the network. Think about how the network would behave if every word in the network has two inputs rather than one. Think about how the network would react if some of the links between words are turned words OFF instead of turning them ON. Now try to imagine how a network of 20 words would perform.

At this point, you should be thinking: these small networks are all well and good, but a real vocabulary network is going to contain hundreds of words, not just a handful, and you should be wondering whether size makes a difference. The next sections are designed to let you explore some bigger networks with some extra additional properties. These networks are still a long way off from being plausible models for real vocabularies, but they will prepare you for working with much bigger networks in the later parts of the workshop.

### 3.3 Some Bigger Networks

We start off by defining some basic properties for these bigger networks:

1. *The networks in this section all consist of 100 words.*
2. *Each word has two inputs from other words in the network.*
3: Each word can be ON or OFF.
4: There are two types of words:
   Some words go ON if ONLY ONE of their inputs is ON.
   Some words go ON only if BOTH of their inputs are ON.

These properties need some explanation:

The size of the network is basically an arbitrary choice designed to make the visualizations easy to follow and easy to understand. We will look at bigger, more realistic vocabularies in the later sections of this workshop. The question you need to be asking yourself at this point is whether the behaviors we get from a small network will scale up. Would we expect a 1,000-word vocabulary perform in a fundamentally different way from a 100-word vocabulary?

The other three properties define the fundamental characteristics of the words that form the network.

Each of the words in the network have two inputs from other words. This is a step up from the networks that we have looked at so far in this workshop, where most words had only one input. The obvious question to ask here is: why have we limited the number of inputs to two rather than three or four, or more? The answer is that we have done this in the interest of keeping things as simple as possible. We could look at more complex networks where each word has many inputs, and we could look at networks where the number of inputs a word has is allowed to vary—some words might have only one connection, others might have five or six, and a few might have 20 or 30 connections. For the moment, this is a level of complexity that we want to avoid, so we will work with models where each word has just two inputs in the rest of this workshop.

Each of the words in the network can either be ON or OFF. Here too you might be wondering why we are limiting the words to only two activity states in this way. Again, the answer is that we are trying to make our models as simple as possible. We could have invented a more complex scale of activation—for example, we could have eight levels of activation, rather than two—or we could have an activation continuum (language teachers often talk about the active or passive vocabulary continuum). However, if we adopted one of these solutions, then we would have to be explicit about each level of the activity scale or each degree of the continuum. Again, that is a degree of complexity that we will avoid for the moment.

There are two types of words. This third characteristic is a significant change from the network models we have looked at so far, where all the words in the network responded in the same way to their inputs. When words have two inputs, rather than one, we have four different input patterns, and words can respond to these different patterns in a number of different ways. With two inputs (e.g., 0 and 1), we have $2^2 = 4$ different input patterns (00, 01, 10, 11), and there are 16 ($2^4$) different ways for any individual word to respond to these inputs. Figure 6 shows the four input patterns and the 16 possible response patterns.

The two columns at the left of Figure 6 show the four possible input patterns. We have two inputs, and each of them can either be OFF (shown as 0) or ON (shown as 1). Each of the 16 columns in the right-hand section of the Figure shows
one possible set of responses a word could make to these inputs. For example, Column A shows a response pattern where the responding word does nothing: whatever inputs the word receives, it responds by obstinately remaining OFF. In contrast, Column G shows a different pattern of responding. A word of this type goes ON if only one of its inputs is ON, but remains OFF when it receives no input or when both of its inputs are ON. Column I identifies a word that goes ON when both its inputs are OFF, but goes OFF if any of its inputs are ON.

The complexity of these patterns is one reason why the models we will be looking at in this workshop only have two inputs for each word. If we allowed our words to have three inputs, then we would have to think about eight different input patterns, and there would be 32 different ways for a word to respond to these inputs. This is far more than we can handle for the moment. So again, in the interests of simplicity, we will choose to work with just two of the response patterns in Figure 6: Pattern B and Pattern H. Pattern B identifies a word that goes ON only when both of its inputs are activated. Words of this type are conventionally called AND words: they go ON if input 1 and input 2 are ON. Pattern H identifies a word that turns ON if any of its inputs are ON. Words of this type are usually called OR words (they go ON if input 1 or input 2 or both inputs are ON). We could use the other response patterns, but for the moment at least, we will stick with just these two. The main reason for this is that it is easy to find a real-world analog for these two response types: Type B words are words that are not easily activated, while type H words are words that activate more easily.

With these basic concepts, we are now in a position to start exploring what a bigger vocabulary network might do. In order to do this, you will need to go to the workshop website: https://www.log nostalgics.co.uk/Workshop/index.htm

This site contains all the programs that we will be using in this workshop. These programs are designed to run in the Firefox browser, and they may not work correctly in other browsers. You will also need a fairly large screen to make them work properly—do not try this on a mobile phone, or a small tablet.

From the Home Page, click on the button labelled Program-1 Basic networks and their properties to access the first set of simulations. This program allows you
to explore how a large number of networks with the characteristics described above actually perform.

Figure 7 shows the start page for the first simulation set in this workshop. This page lets you vary the values of two parameters, NTWK (network) and INIT (initialization) that define a network. NTWK sets up a 100-word network by deciding how words are connected to other words, and by deciding how each word responds to its inputs. Varying the values of the NTWK parameter will give you a network where the words are connected together in a different way (still with two inputs each), and will also change the way individual words respond to their inputs. INIT sets about half of the words in the network ON. Varying the value of the INIT parameter will set a different selection of words to ON when the program starts up. Leave these parameter values alone for the moment, and just click the SUBMIT button. The program should return an output that looks something like Figure 8.

Figure 8 shows how one instance of a 100-word vocabulary behaves. The top line of the diagram shows the current state of each of the words in the vocabulary after the network has been initialized at random. Dark squares show words that are ON, light squares show words that are OFF. Initially, about 50 words are turned ON. These words send signals to the words they are connected with, as described earlier. The connected words update their activity status in response to these signals, and the network moves to a new state which is shown in line 2 of Figure 8. So, in line 1, Word6 is OFF, but changes status as a result of the inputs it receives from other words, and in line 2, Word6 is ON. Similarly, Word8 is ON at time 1, but is turned OFF at time 2, only to turn back ON again in line 3, which shows the pattern of activation when line 2 is updated. And so on. Figure 8 shows that the number of activated words drops steadily with each successive update. After 9 updates (line 10), only 13 words remain activated. After 14 updates (line 15), only 1 word remains activated. After 15 updates (line 16), the network has reached an attractor state where all the words are OFF and there is no remaining activity in the network.
However, not all networks behave in this way. You can examine the performance of other networks by changing the value of the NTWK parameter in the Program-1 start page. Changing this parameter gives you a different network: the new network will still contain 100 words, but the words will have different inputs and the way they respond to these inputs will be different. In these first simulations, you do not have any direct control over these characteristics, but each different value of the NTWK parameter will give you a different network.

Figure 9 shows the performance of NTWK 1235. Like the network shown in Figure 8, NTWK 1235 settles into a steady state after a small number of updates, but here, while most of the words in the network have turned themselves OFF, a small number of words seem to be frozen in the ON state.

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

- Why do some networks produce words that are permanently ON?
- How common are networks that do this?
- Do we ever get a network where ALL the items are permanently ON?
- What conditions are necessary for this to occur?
- Can we move a stable network out of its attractor state?
- What conditions would be necessary for a new attractor state to appear?

You can start to explore ideas like this by playing with the Program-1 and varying the two parameter values on the start page. The NTWK parameter determines how the vocabulary model is set up—how the words are interconnected, and how they react to their inputs. The INIT parameter only changes the initial values that the words in the model are assigned. So, if you run a set of simulations with
NTWK model 1234 you will always get the same model, but you can vary the initial randomization of the words by using different values of the INIT parameter. Similarly, if you fix the value of the INIT parameter, you can vary the value of the NTWK parameter, and see how different models respond to the same initialization. You will probably want to run 20 or 30 simulations in this task to get a feel for how these first models work, and how many different types of performance you can identify. For instance, you could run NTWK 1000 with five or six different values for the INIT parameter (1000, 1001, 1002, 1003, 1004 and so on). Do these different values change the way NTWK 1000 behaves?

There are literally thousands and thousands of different ways to set up a 100-word vocabulary network, even when you have only two characteristics that vary from one network to the next, so it is very unlikely that you will find two networks that produce exactly the same results. To help you keep track of interesting models, it is a good idea to start a lab notebook to keep a record of how your simulations are performing. This is not strictly necessary for the simulations using Program-1, but it will be increasingly useful as the workshop models get more complicated. For each simulation that you run, record the parameter settings that you use, and make a note of any interesting features that you notice in the program’s report page.

You should find that every different network produces a different outcome. Most of the networks will reach a stable attractor state pretty quickly. All networks with the characteristics described above have at least two attractor states—one where all the words are ON, and one where all the words are OFF. However, most networks end up in an intermediate attractor state where some words are ON, and the rest are OFF. We will look at the implications of this in more detail in part two of the workshop. For the moment, it is worth noting that,
surprisingly, changing which words are ON and which words are OFF when a network is first initialized does generally not result in the network moving into a new attractor state. You can explore this idea for yourself by running Program-1 using different values of the INIT parameter. You should find that changing this value will sometimes result in a network finding a different attractor state, but this does not happen very often. Networks of this size (100 words) typically have only a small number of attractor states. Again, this has some implications for the behavior of real vocabulary networks that we will explore further in Workshop 2.

3.4 Discussion

We began this introduction with a very small vocabulary network which contained only five words. Of course, we are not suggesting that a network as small as this could ever hope to be a good model of a real vocabulary: it is when we start to work with larger networks that some interesting properties begin to emerge that may be relevant to real vocabularies. The most important of these features is that these larger vocabularies are much more stable than we might have expected. Each of the models you worked with in Program-1 were made up of 100 words which could either be ON or OFF, so in theory there are \(2^{100}\) possible states for the network to take up when it is randomly initialized. This is a colossal number of different possible arrangements, and yet after only a handful of network updates a self-reinforcing attractor state always emerges.

Taking all this into account, the suggestion that we want to explore in this workshop is that small network models might be able to act as *vehicles* for an exploration of real vocabularies. Simplified though they are, these models show complex behavior patterns which might throw some interesting light on the way real vocabularies behave.

Let us begin with some simple observations based on the networks we examined in Program-1. These networks have some obvious resonances with real vocabulary networks. It is “obvious” (we will come back to this later!) that real vocabularies consist of a collection of elements that are connected together in some sort of network. It is reasonable to assume that the elements that make up a vocabulary are the words that the owner of the vocabulary “knows.” It is also reasonable to assume that these words are “connected” in some way, though we do not know how exactly words in real vocabularies are linked, or how many connections we are dealing with. Clearly, the simple models that we have looked at here do not fully capture this complexity. Indeed, we have made some massive simplifications here: we have not defined what we mean by “a word” or what we mean by “knowing” (recognizing the word, knowing how to use it? cf. Richards, 1976 and Nation, 2001). Nor have we said anything about meaning. But if we suspend our disbelief over these issues for the moment, we can see that simple network models can look quite plausible as analogs of a real L2 vocabulary.

More importantly, accepting this analogy at face value immediately gives us a number of free gifts. The main gift is that it throws some interesting light on the issue of active or passive vocabulary in the mental lexicons of L2 learners, one of
the big theoretical problems in L2 vocabulary acquisition research (cf. for example Melka Teichroew, 1982, 1989 and Melka, 1997).

There is a huge literature dealing with different evaluations of active and passive vocabulary. It is generally agreed that passive vocabulary consists of words that you can recognize but would not normally be able to use without some form of prompting, while active vocabulary consists of words that you can recognize AND use. It is widely taken for granted that a large active vocabulary is a Good Thing. It is also widely assumed that all words start off as part of a learner's passive vocabulary and that some of these words eventually end up as active vocabulary items. Furthermore, much research refers to a passive or active vocabulary continuum, where words start off at one end of the continuum and gradually move toward the other end. The question is what drives this change? What makes a passive word permanently active? and how does this change happen? There are no good answers to these questions, partly because the properties of the “continuum” are rarely well-defined in this research, and partly because it is rare to find discussion of any sort of mechanism which would push a word along the continuum. Our model networks begin to suggest a rather different approach to this problem.

To start, it seems reasonable to identify active vocabulary in a real vocabulary network with words that are ON when a vocabulary network is in its stable attractor state. These are words that can be used even in the absence of an external stimulus. We can also identify passive vocabulary as words in a network that are OFF, but can become active when some sort of external stimulus is applied to the network. For example, we could activate an inactive word directly, or we could activate other words in the network that have the effect of causing a passive word to become active (for a while). This is essentially what happens in real life when learners read words: they can often recognize words that appear in a text even if they cannot use them without this supportive context. Note that we do not have to build the idea of active and passive vocabulary into our model as a fundamental assumption: this key idea just appears as an emergent property of a vocabulary network where words have two possible activation states.

We could, of course, design a model vocabulary which has more than two activation states—but it is traditional in simulation studies to keep things as simple as we possibly can until we are forced to make them more complex. In any case, we are interested in the idea that what looks like complex behavior in a vocabulary might actually be generated by interactions between simple components, and it is useful to keep these components as simple as we possibly can. Therefore, for the moment we will stick with a binary passive or active vocabulary.

In fact, the simulations suggest that some vocabulary networks might contain more than two types of word. In some vocabulary networks, we find words which are unstable, sometimes ON and sometimes OFF, sometimes activated, sometimes deactivated in successive updates of their network. This situation arises when a network’s attractor state is actually a short cycle of states rather than a single steady state. You almost certainly found some examples of this when you worked with Program-I on the website. These words, flickering in and out of the active state, seem to make up a significant proportion of any learner’s
vocabulary, though the existence of intermittently active words is rarely discussed in the research literature. Our model vocabularies do not just draw our attention to the appearance of this phenomenon: they also provide a natural explanation for why it occurs, and suggest some limitations on the number of intermittently active words we would expect to find in a vocabulary.

And maybe this prompted you to ask:

*Why do some words oscillate between the ON and the OFF state?*

*What conditions are necessary for oscillating words to appear?*

*How big is a typical set of oscillating words?*

*In a growing vocabulary, do all words start off oscillating between ON and OFF?*

*How could we examine the appearance of oscillating words in real life?*

*What changes would you need to make to a network to make an oscillating word stable?*

*Do we get words that oscillate together?*

*Do we get words that oscillate in complementary pairs?*

*What other structures are likely to emerge in a vocabulary network?*

This set of questions is a good example of the way working with models suggests further research, but you should note that some of these questions are very different from the questions that vocabulary researchers have traditionally asked.

Furthermore, the model networks hint that there might be a fourth kind of word. Some OFF words seem to become active as a result of activation by other words in the network (low frequency synonyms, perhaps?), while other words have a very low likelihood of being activated in this way. This sort of situation is likely to arise if the network includes words whose current state depends on a chain of other words which are usually OFF. The chances of one of these words being activated will normally be vanishingly small, especially if the network itself is large, but they will quickly become activated if other words that they depend on are activated. This idea might lead you to ask questions like:

*How many other words are typically activated when an arbitrary word is turned ON?*

*Do networks typically contain words that cannot be turned ON by turning other words ON?*

*What conditions would be necessary for this to be the case?*

*What would be necessary to turn these words into active vocabulary items?*

*What other type of chain effects would you expect to find in a network?*
Considerations like these suggest that our simple two state model is actually much richer than it appears at first sight. Not only does it give us a very simple explanation for why some vocabulary is active while other words remain passive, it also suggests that words might naturally fall into other types as well—types which do not arise naturally in the active or passive continuum approach. In this way, the model vocabulary networks highlight a significant gap in our normal thinking about L2 words. More importantly, they provide a relatively simple and straightforward mechanism as to why these different types of words appear in real life: different word types are just an emergent feature of the vocabulary network that the words belong to. The models strongly suggest that it might be wrong for us to think about active or passive status as an intrinsic property of a word. Rather the current status of any individual word is actually a property of the network as a whole, and not a fundamental property of the word itself: if we change the structure of the network by altering the way a word is connected to other words in the network, or by changing the way a word responds to these inputs, then the current activity state of the word is also likely to change, and the activity status of other words is likely to change with it (cf. de Saussure, 1916 for an early version of this idea).

In addition, these simple models suggest that apparently obvious questions about active and passive vocabulary might not be as straightforward as they appear to be at first sight. A simple binary distinction between active and passive vocabulary is already richer than we might have anticipated, and might prompt us to ask whether we really need to have a complex theoretical construct such as the active or passive continuum. This leads on to other questions:

- **Does it make sense to ask whether the development of active and passive vocabulary is “the same or different”?**
- **Is vocabulary acquisition a single process that just has two different outcomes?**
- **Does it make sense to ask about “the relationship between active and passive vocabulary”? Why would we assume that there is any relationship of this sort?**
- **Does it make sense to ask whether a learner’s active vocabulary is always X% smaller than their passive vocabulary?**

On the other hand, the simple models lead us to ask some rather different questions:

- **How active is an ideal vocabulary?**
- **Is a network where all the words are active “better” or “worse” than one where the level of activity is lower?**
- **What criteria would you use to decide this question?**
- **Is a potentially active vocabulary—a vocabulary which can quickly switch itself into an active state—an efficient solution to the problem of vocabulary storage?**

Questions of this sort arise naturally when you run simulations, but they are not questions that can easily be addressed when your main mode of research is large...
scale experimental studies, difficult to organize, and often fraught with ethical issues. Working with models allows you to start thinking about the implications of these questions without the huge investment of time and resources that accompany real-life experimental studies.

A second gift that arises naturally in a model vocabulary network is that we find chains of words which are dependent on each other—sets of words which become active only if a particular source word is activated. We will examine this idea in more detail in a later section too. For a moment, let us just note that this might provide a way for words in a vocabulary to naturally form themselves into semantic or thematic clusters—sets of words which are activated together in response to a common source word.

The interesting thing here is that working with even a grossly simplified model vocabulary is already making us think about real vocabularies in a new way. This is going to be a recurrent theme throughout this workshop, and we hope that working through the simulations will fundamentally change the way that you think about vocabularies. For the moment, though, it is enough to note that a relatively superficial consideration of some very simple models of the way vocabularies work has already led us to consider the role of some of the basic constructs in L2 vocabulary research. This is fairly typical of what happens when you work with simulations, and one of the main strengths of this type of work.

Finally, an important idea that runs through this workshop is that working with simulations may appear at first sight to be an easier option than running standard experimental studies, but actually this type of work is not as easy as it looks at first sight. In traditional research, you often do not need to be very specific about how your theory of vocabulary acquisition actually works—it is sufficient to point the reader in the general direction of your thinking without providing much detail. It is also difficult for critical readers to prove that your theoretical model is at fault. This explains why we get so many contradictory results in the research, and why we find so many weird and wonderful metaphors in the research literature. Simulation research does not let you get away with imprecise thinking of this sort. Unless you are absolutely specific about what your simulation has to do, then it just will not work. For instance, you cannot just say “under X conditions words are learned” without specifying exactly what the conditions are, and exactly what you mean by “learned.” Furthermore, when you break down a large concept like “learned” into its component steps, you often find that the order in which instructions in a simulation are implemented can sometimes make a huge difference to the way a model runs. Sometimes this forces you to rethink what you are modeling, or it makes you realize that a feature you thought was marginal to your model is actually a crucial feature instead. Or again, you sometimes find that your model only works within a certain range of parameter values. Always, doing simulations means that you have to be absolutely explicit about the assumptions you are making, and this means that your thinking is cruelly exposed to critics. Generally, this is a good thing, though it can sometimes be a bit scary. Simulation research is not an easy option, but it makes you think in a way that traditional research approaches often do not, and the results are often exhilarating and exciting.
The next set of simulations, Part 2 of this workshop, will begin to examine some of the ways we think about L2 vocabularies, and will give you some first-hand experience of just how unsettling working with simulations can be.

Acknowledgements
The authors would like to thank the editors and the two anonymous reviewers for their comments and feedback.

References


Vocabulary Networks Workshop 2: Activating Words in a Network

Paul Meara\textsuperscript{a} and Imma Miralpeix\textsuperscript{b}
\textsuperscript{a}Swansea University; \textsuperscript{b}Universitat de Barcelona

Abstract

Workshop 2 explores some larger vocabulary networks than those presented in Workshop 1 (1,000 words instead of 100), and introduces some additional ways of activating words in a network (e.g., by using new parameters in our models). The simulations will show that raising a network’s activation level may not be as easy as we would have expected, and that different kinds of events may cause either temporary or permanent changes in the networks. Findings will be discussed in relation to the previous literature on passive and active vocabularies. Several implications for vocabulary learning and teaching will also be considered.

Keywords: Active vocabulary, Attractor states, Network structures, Simulations, Vocabulary modeling

1 Introduction

In the first of these workshops (Meara & Miralpeix, 2022), we looked at some very simple models of a vocabulary network. The models were small—only 100 words—and the words they contained had only a minimal set of properties. Each word was linked to two other words, and each word responded to input from these linking words in a fixed way. The simulations showed that a simple network of this sort behaves in a way that is more complex than we might have predicted, and we saw that some interesting emergent features are beginning to appear even in these tiny models. In particular, we noted that the networks naturally and quickly move into a stable attractor state. We also noted that whether a word can be described as active or passive appears to be a property of the network as a whole, rather than an intrinsic characteristic of the word itself. In this workshop, we will explore these ideas in more detail.

2 Some Bigger Models

The models that we use in this section are essentially the same as the models we used in Part 1 of the Workshop, but they are much bigger. Here, the models consist of 1,000 words, rather than 100 — 10 times bigger than the networks we looked at in Part 1. This is still small in comparison with a real vocabulary which might contain several thousand words, but the networks in Workshop 2 are not totally unrealistic: many second language (L2) learners have vocabularies about this size (cf. Nation, 1996 where the term “Little Language” is used to refer to the
first thousand words learners know, sufficient to make themselves understood and to perform a series of basic tasks). This means that we are beginning to look at vocabularies on a realistic scale.

In this workshop, we will continue using the programs we used in Workshop 1, so the basic approach will already be familiar to you. However, the bigger size of the vocabularies (1,000 words instead of 100) means that we need to change the way the programs display the information that they generate. Accordingly, we will start off by looking at the new display format, so that you can familiarize yourself with how it works.

To begin, you can go to the workshop home page: https://www.lognostics.co.uk/Workshop/ and click on the button labelled Program-2 Activating words. The control panel for this program is shown in Figure 1.

This control panel is basically the same as the control panel in the previous Workshop, but here you have five variable parameters instead of two. The first two parameters, NTWK (network) and INIT (initialization), work in exactly the same way as they did in Part 1. NTWK controls the basic structure of the network: it determines the links between the words, and the way each word responds to its inputs. INIT determines the random activation pattern that the network starts out in. There are also three new parameters, all concerned with events that you can control. They allow you to make something happen to a network that has settled into a stable attractor state.

The next step is to run some simulations (you can run these simulations yourself) with the three new parameters set to 0. You do this by clicking the SUBMIT button. The program will return a report that looks something like Figure 2.

This new report screen shows you the number of active words in the network each time it updates itself, but it does not show the status of individual words—1,000 words is just too many for us to show them individually. We will be using lots of outputs like this throughout this Workshop, so it is important that you are able to interpret outputs like this one. In Figure 2, the network being examined is NTWK 2711. It is made up of 1,000 words. The number of active words in the
network is indicated on the Y-axis by a line of green dots. The X-axis shows how this value varies over time as the network updates itself. Figure 2 shows that you have 1,000 updates in the simulation. In this report, the number of active words starts out at about 50% (recall that we initially set this value randomly, with about half of the words ON and half of the words OFF). Then the network adjusts its level of activation in response to this initial randomization. As the network updates itself, the number of activated words rises for a while, then falls back again, until after roughly 50 updates the network reaches an attractor state where about 620 words are activated. Once it reaches this attractor state, the network stays there. Experiment with different values for the NTWK parameter and the INIT parameter, and make sure that you understand what the new format is telling you.

Now let us build a simple real-world scenario around the data in Figure 2. Suppose we are dealing with a beginning learner of English—call her Maria. Maria, as a beginner learner, has a passive vocabulary of 1,000 words, but her active vocabulary is limited—only 620 words. The other words in her vocabulary—380 of them—are words that she knows only passively: these are words that she would recognize, but she is not able to produce them without help. Most research agrees that passive vocabularies tend to be bigger than active vocabularies, although there is no agreement on how large the possible gap between the two could be (e.g., see Melka, 1997 or Miralpeix, 2020 for a review), so these figures are arbitrary but plausible as a first stab. Obviously, this is not where we want Maria to be: we would really like to increase her productive vocabulary. Ideally, perhaps, we might like all 1,000 words to be part of Maria’s productive vocabulary. The question we need to ask, then, is what do we need to do to the network in order
to achieve this outcome? The obvious answer is that we might be able to increase the number of active words in Maria’s vocabulary just by turning more words ON, and hoping that doing so will result in a permanent increase in her active vocabulary size. Program-2 is designed to simulate this process.

You can go back now to the start page for Program-2. In addition to the two standard parameters, this start page includes three new parameters, nEv, sEv, and rEv. These three parameters allow you to control events that take place when you run your simulations. In Program-2, an event turns a number of randomly chosen words ON, and shows you what effect this has on the overall level of activity in the network.

- nEv sets the number of events you want to take place. For the moment, set the value of nEv to 1. This means that when the simulation runs you will get one event. However, you cannot control exactly when these events will take place, as the program decides this randomly.
- sEv sets the size of any events. This parameter controls how many words will be turned ON whenever an event occurs. Set the value of this parameter to 10. This means that each time an event takes place, 10 additional words will have their current value set to ON.
- rEv controls which words are affected by an event. You cannot control the fine detail of this in Program-2. For the moment, it is sufficient to know that specific values of rEv will always affect the same words each time you run the program. Set the value of rEv to 1234.

You can run the simulation by clicking on the submit button. You should get an output that looks something like Figure 3.

![Figure 3. Program-2 Report page. NTWK: 2711, INT: 1234, nEv: 1, sEv: 10, rEv: 1234.](image-url)
Figure 3 tells us that this network starts off with about 500 words in their activated state and, after a small number of updates, the network settles into a stable attractor state. The red dot at the bottom of the figure tells us that a single event takes place around update 690. The effect of this event is to momentarily increase the number of activated words in Maria’s vocabulary by 10 words. However, this burst of activation does not last very long—a small number of further updates brings the overall level of activation in the network back to its attractor state. So far, it looks as though just turning 10 words ON once does not have any long-term effect on Maria’s vocabulary.

3 Program-2: Raising the Activation Level of a Stable Network

From here, you should be able to see that we can experiment with different combinations of the parameters, and examine the effects of turning words ON in different ways. You will perhaps be thinking of a number of obvious lines of enquiry and we have listed some of these below. The questions we suggest in this section do not have any “right answers.” They are primarily designed to make you think about what the simulations are doing, and to suggest further questions that you might ask. What you should notice is that experimenting with different models makes you think about vocabulary networks in new ways. Here are some starter questions:

Are larger events more likely to have a permanent effect on the number of active words in a vocabulary network? You can test this idea by changing the value of the sEv parameter. For example, if you set sEv to 50, then the program will turn ON 50 words each time an event occurs instead of 10. How does NTWK 2711 react to a single event of this size? Increase the value of sEv and watch how this change makes a difference to the way this network reacts.

Do all networks react in the same way as NTWK 2711? You can examine this idea by varying the value of the NTWK parameter. Choose a set of values that give you networks with different attractor states. How often do you find a value of sEv that results in a permanent change in the overall activation level of a network? Can you identify some networks that do not react like NTWK 2711? Why do you think this might happen?

Do frequent small events have a permanent effect on the number of active words in a vocabulary network? You can test this idea by changing the value of the nEv parameter. For example, if you set nEv to 50 then the program will give you 50 events instead of one. Most often, you will find that simulations with repeated events look something like Figure 4. This simulation shows the same network that we saw in Figure 3, but with an increased value for the nEv parameter. In Figure 4, nEv has been set to 25, giving us 25 events, each activating 10 words. Again, events of this size and frequency have a small effect on the overall activation of words in the network. Each event causes a small ripple of activation in the network, but these changes are not permanent, and when the events stop around update 900, the network quickly returns to its attractor state.

Program-2 does not give you any direct control over when the events occur. Usually, the events will be spaced out more or less evenly across the X-axis of the
report page. However, if you experiment with different values of the rEv parameter, you will come across patterns of events that look more interesting. Sometimes you will get a cluster of events that happen in quick succession, and you might find that this causes more activation in the vocabulary network. This should lead you to ask:

**Do clusters of small events have a permanent effect on the number of active words in a vocabulary network?** You can test this idea by choosing a value of rEv that generates clusters of events. Generally speaking, you should find that small numbers of events which activate a small number of words have only a limited effect on the overall activation level of the vocabulary network, but when you increase the number of events and increase the size of each event, the overall activation levels will increase and be longer lasting. Sometimes you will find that the network will jump to a new, higher level attractor state, but occurrences of this sort are relatively rare.

You should find that there is a considerable amount of variation in the way the networks respond to different types of events. Figure 5 shows an example of how one network with a low level of activation behaves in response to a series of large input events. NTWK 1946 has a very small number of active words when it is in its attractor state, so we might expect it to be particularly responsive to events that activate large numbers of words. Think about how you would expect this network to behave before you go on to run the simulation yourself.

Here, we have asked for 20 events, and in each event 100 words are turned ON. The result is a huge amount of activity in the network, particularly when we have a cluster of events occurring in quick succession, such as the cluster around

*Figure 4. NTWK: 2711, INIT: 1234, nEv: 25, sEv: 10, rEv: 1234.*
update 400. But surprisingly, none of these events seems to result in a permanent increase in the number of active words. This particular network stubbornly resists being shifted to a higher level of activation. Is this what you predicted? Why not? In fact, even if we ask for huge events, where 500 words are activated each time an event occurs, the long-term effect for this network appears to be negligible (cf. Figure 6).

However, not all models are as stubbornly resistant to change as this one is. Figure 7 shows an example of a model where a small number of input events does have a long-term effect on the overall activity level of the network. This figure shows a network that has an attractor state which already has a large number of active words. Each time an event occurs, the network moves to a slightly higher attractor state, and it seems to stay there.

This might prompt you to ask: are highly active vocabulary networks more likely to respond to activation events than networks with low activity levels? You can investigate this question by looking at a range of models with different attractor states. We recommend that you find a set of 10 networks with an attractor state with a lot of active words, 10 networks with an attractor state where about half the words are active, and 10 networks with an attractor state where only a few words are active. Does the level of activation in a network affect the way it responds to further activation events?

At this point in particular, you should be asking whether Program-2 is really a good model of how a vocabulary might become more active. The answer to this is that it probably is not a good model, but it is useful to think about why not. You
Figure 6. The Effect of Very Large Events (Each Event Turns ON 500 words). NTWK: 1946, INIT: 1234, nEv:20, sEv: 500, rEv: 1234.

Figure 7. How a High Activation Network Responds to Large Input Events. NTWK: 1234, INIT: 1234, nEv: 10, sEv: 50, rEv: 500.
should also be thinking about what additional features would we need to build into the models to make them more realistic. Despite their lack of realism, are the models in this section perhaps drawing our attention to features of real vocabularies that our current theories do not take account of?

You should also be asking whether the models in this section are giving us the information that we really need. For example, is the total number of activated words in a vocabulary network the best way of assessing its performance? Maybe a different measure would be more useful? What might that be? When an event activates a large number of words, the overall activation level of the network rises instantly, but it fades away more slowly. This implies that there is a lot of underlying activity in the network that the report is not picking up on. Maybe, instead of just counting the number of active words at each update, we could look at how many words are changing their current state at each update of the network? Would this perhaps give us more insights into how the network is responding to the activation events?

More seriously, we might also ask whether it really makes sense to try to shift a network directly from one attractor state to another, just by activating a few words. All the simulations in this section have started off in one of their attractor states, and your explorations with Program-2 should have convinced you that it is surprisingly difficult to shift a network out of an attractor state into a new one just by turning words ON. However, it is possible that Program-2 is approaching this problem in the wrong way. Maybe vocabulary networks are not normally in an attractor state when events happen? We will look at this possibility with Program-3.

4 Program-3: Networks with a High Level of Temporary Activation

Program-3 lets you examine some of the more speculative ideas in this discussion. So far, we have only considered how we might shift a vocabulary network from one stable attractor state to another stable attractor state where more of its words are active. However, it is possible that networks in an attractor state are particularly resistant to change, and it is possible that a network that has been shifted out of its attractor state into an unstable state (an excited state) might be easier to manipulate. Basically, we need to explore whether an excited network can be prevented from falling back into its attractor state by activation events. Specifically, Program-3 lets you ask: what happens in a vocabulary network that has been shifted out of its attractor state BEFORE any events to take place are allowed to take place. Normally a network that has been shifted out of its attractor state would return to its attractor state after a small number of updates. However, it is plausible to imagine that a network that is in an excited state might respond differently. Perhaps randomly turning words ON in an excited network could prevent this return to normality, and bring the network to a different, higher attractor state? How effective would we expect this to be?

You can go now to the Workshop homepage and click on the Program-3 button. The start page for this program is shown in Figure 8: this page is similar to the start page for Program-2, but it contains an additional parameter that you can experiment with: **SPK (Spike)**. The simulations in Program-3 run in exactly

---

*Searle and Mark: Workshop 2: Activating words in a network*
the same way as the earlier simulations in this section, but parameter SPK allows you to apply a single large spike of activity to the network, as well as a series of smaller input events.

This spike always occurs at update 99—the network will normally have settled into its stable attractor state by this time, so if you set the SPK parameter to a high value, it will jolt the network out of its steady state into a state which is more unstable. Maybe a small amount of input will be enough to keep the network in this state of heightened activity?

You can now set the NTWK parameter and the INIT parameter to 1003, nEv to 250, sEv to 200, rEv to 1003, and SPK to 500. This combination of parameters will give you a network which has a low-level attractor state (190 words are ON). The way this network behaves is illustrated in Figure 9.

This figure shows NTWK 1003 moving to its normal attractor state—just short of 200 active words (updates 0-99). At update 99 the program turns 500 words ON, and this results in a sudden uplift in the network’s activation level. After just a few further updates, this figure has risen to just over 900 words: nearly every word in the network is now ON. Then, following this initial spike of activation, a series of events takes place, where each event turns ON 200 words. There are 250 events of this sort, but even when the events are as frequent and as large as this, the overall activation level of the network does not appear to be permanently affected. As long as the activation events continue, the number of active words in the vocabulary remains high. However, when the activation events stop at update 900, the network returns almost immediately to a low level of activity. Interestingly, this new attractor state is not exactly the same as the one this network started out with. The new attractor state has a few additional words that are permanently activated as a result of the massive amount of stimulation that we
have provided it with. This result suggests that turning words ON does sometimes cause a permanent change to the activity level of a network, but it is not obvious that a change of this kind is directly related to the amount of input that the network receives.

Here again, you should be asking questions about what Figure 9 is telling you. Would we get a bigger permanent uplift if we increased the number and the size of the events in this simulation? (Change the values of nEv and sEv to explore this question). Would we still get the uplift recorded in Figure 9 if we reduced the number of events, or if we reduced the size of the events? Does Program-3 produce simulations which exhibit threshold effects where a small change in the number of events or the size of an event generates a sudden change in the way the network behaves? Is the initial spike at update 99 really important, or is it just the combination of very frequent and very large events that gives us the pattern of activity reported in Figure 9? Do all networks with low levels of activity behave like this one? Do networks with a naturally high attractor state react in the same way as NTWK 1003? You can investigate all these questions by using Program-3 with different values for the parameters.

5 Discussion

Four main ideas seem to be emerging from the simulations in this section. First of all, the main feature that you can expect your simulations to show is that it is surprisingly difficult to shift a vocabulary network from one attractor level to different higher-level attractor. As we have seen with Program 2, a network in
an attractor state will usually just absorb small changes and go back to its stable resting state. The main exception to this is when your simulation implements a lot of events, and these events are large enough that they overwhelm the network’s natural tendency to go back to its resting state. In this case, you can sometimes find that very high levels of activity can be supported by the network. However, cases where this happens appear to be rare. In most cases, a raised level of activity is only temporary. When the activation events stop, then the network will most often fall back to its stable resting state, even when the number of activated words has been very high.

What are the implications of this? It is difficult to draw any immediate conclusions from these simulations, because the parallels between the simulations and real life are a bit tenuous. We cannot turn passive words ON in real life—all we can do is provide some conditions in which some words might get turned ON, by getting students to read, for example, or sending them off on an intensive residential course. The simulations do not mirror treatments of this kind directly. Nevertheless, the inescapable conclusion seems to be that activities for learners that just turn passive vocabulary items ON temporarily are not in themselves going to be sufficient to increase students’ active vocabulary on a permanent basis, and this does have some real-world implications. As Nation (1990, p. 94) noted, “if learners have a reasonably large receptive vocabulary but are unable to put enough of this into productive use, then the teacher needs to concentrate on activities that enrich the learning of known words and improve access to them.”

Surprisingly, quite a lot of standard vocabulary exercises could be seen as events which turn passive words ON (see Meara, 1990). Consider, for example, an exercise like the one shown in Figure 10.

Suppose that our learner—Maria—does this exercise, as part of an activity designed to improve her vocabulary. If COMB is already part of Maria’s active vocabulary, then this exercise just allows her to use a word she already knows: in the simulation, this would be equivalent to turning ON a word that is ON already, and we would not expect such an event to make any difference. If COMB is part of Maria’s passive vocabulary, then she will recognize the word when she sees it, it will become temporarily active, and she will be able to use it correctly to fill the gap in the example. In the simulation, this would be the equivalent of turning ON a word that is currently OFF. But once the network is updated, it is unlikely that COMB will remain active: it is much more likely to revert to its normal passive state. The simulations in this chapter suggest, therefore, fill-in-the-gaps exercises like this may not be an effective way of making COMB an active vocabulary item for Maria.

Which word best fits the gap?

He is using a ____ to arrange his hair

1: razor          2: comb          3: tissue          4: toothbrush

Figure 10. A Typical Vocabulary Exercise.
This does not necessarily mean that exercises of the type illustrated in Figure 10 are no good. If you have used these exercises with your students in the past and found them effective, then a simple simulation like this one is not going to convince you that the exercise is not theoretically motivated and should therefore be abandoned. What should happen is that you realize this exercise is not working the way you think it is. The simulations seem to suggest that just activating a passive vocabulary item for an exercise is not enough to turn it from a Passive word into a permanently Active word, so we need to ask what does an exercise of this type do? Maybe the exercise works because it somehow makes COMB easier to activate, or maybe the juxtaposition of RAZOR, COMB, TISSUE, and TOOTHBRUSH changes the connections that COMB has with other words in the vocabulary network. These are ideas that we will explore in the next section of this workshop. What the simulations seem to confirm, though, is that this type of practice will not suffice to improve productive vocabulary use in tasks. In input-driven narratives, for example, where the speaker is pressured to be precise and find the exact wording (Skehan, 2009), words activated with exercises similar to that in Figure 10 may not be available, as they may not have become permanently active.

A second feature of these simulations is that large events which occur frequently are more likely to deliver a permanent change than small events that take place infrequently. This is not really surprising, perhaps—research has shown that immersion settings can speed up learning when compared to instructional settings, with less intense exposure (Foster, 2009; Milton & Meara, 1995; Zaytseva et al., 2018). Psychological studies have also indicated that an effective learning process is characterized by increasingly spaced repetitions, with a short gap in between early meetings and larger gaps in later meetings (Pimsleur, 1967). The simulations confirm these previous findings. Additionally, studies on reading or TV viewing have proved that retention rates are low if vocabulary is not recycled (Gesa, 2019; Waring & Tataki, 2003).

Note though, that the X-axis of the simulations does not actually specify how much time elapses between each update, so we cannot immediately convert the simulation into real time.

However, we could start to ask some questions which examine this idea. For instance, we might ask: in real life, for how long does a normally passive word remain activated, and for how long does the effect of this activation last? Clearly, if the effects of activation are all dissipated within a few hours, or even a few minutes, then it is going to be very difficult to meet the conditions necessary for a permanent uplift in the number of words in a learner’s active vocabulary. In the simulation sets in this part of the Workshop, we have implemented 1,000 network updates which take place in a sequence, but we have not actually said anything about how often these updates take place. When does a network update take place: every few minutes, every day...? If the updates take place daily, then the 1,000 update events recorded in Figure 3 would take about 3 years to be completed, and this is probably not a realistic time scale. Maybe, then, we should be thinking about network updates that take place more than once a day—perhaps every hour. But this implies that temporary activation of a normally passive word would be very transient indeed—a word might be known now, but forgotten again in an hour’s time. What are the implications of this?
The simulations in this section also assume that the network is updated regularly, but this might not be the case. We could have coded the program so that the network is updated at irregular intervals. Or we could have programed the simulations so that activation events do not always occur with an update event. This should make you think about whether updating the current status of ALL the words in the network simultaneously is the best way to model how a network responds to being stimulated. Maybe the network only partially updates itself? Maybe it only updates itself when it needs to? How would a network know when this was necessary?

Again, there are no obvious answers to these questions. The point of this discussion is to show you how running simulations makes you think about the assumptions that underpin normal practice, and to ask questions about these assumptions. The simulations do not provide immediate answers for these questions, but they do help to identify areas where current theory is underdeveloped, or in need of rethinking. They also provide a rich source of ideas for further research. Some straightforward projects which emerge from this simulation set would include the following questions, which do not typically appear in the current research literature:

In our simulations the vocabulary items can either be ACTIVE or PASSIVE, but a lot of the research literature assumes that this feature is graded rather than binary. Is this important? Would a network where words can have several different levels of activation, or different types of activation, behave very differently from the networks we have looked at here?

If you activate a passive vocabulary item, for how long does it remain active?
Do words differ very much in this respect?
Do different ways of activating passive vocabulary items produce longer or shorter activation periods for individual words?
Does vocabulary size make a difference? That is, does activation last longer if you have a larger vocabulary?
Is activation different for beginning language learners than for more advanced learners?
Is there a reliable method for identifying passive vocabulary items?
Are vocabulary networks normally in their stable activity state, or are they normally in an excited state where activity is greater than we would expect?
Do words in a network spontaneously (and randomly?) activate themselves?
Does spontaneous activation of this sort have a noticeable effect on a network?
How could we investigate this phenomenon in real language learners?

The third feature that should be emerging from your simulations is that the details of the model are more important than you might have expected. We already know that the network models in this section are only very rough approximations to real vocabularies, and it is important to stress that we are not suggesting that network models of this sort are intended to be accurate models of how real vocabularies work. Their main role is to be suggestive and provocative. But the same argument applies to the way we have modelled the events in this simulation set. Working
with simulations forces you to think about whether the way you have modelled things is really sensible.

The basic approach we have adopted in this section is sketched out in Figure 11. Here, we have chosen to set up the network as a set of words which start off with fixed properties that do not change as the program runs. The only characteristic that changes in Program-2 and Program-3 is that some words move from their ON state to their OFF state, and vice versa. Their other properties remain fixed. The simulations do not change the way words are linked to other words, and they do not change the way words respond to other words. But maybe this approach is too rigid, and we should consider what happens in a network where these properties are not fixed? We will look at these issues in Part 3 of the Workshop.

We have also chosen to model networks that are of a fixed size—1,000 words in this section—but this means that we are not looking at dynamic networks which are growing or shrinking in size. We are also assuming that all words have the same characteristics and the same status. Our guess is that neither of these assumptions is correct, and a much safer assumption would be that words are much less homogeneous than we have modelled them. In particular, it seems a good guess that words which have only recently been learned do not perform in the same way as more well-established words. How could we build this factor into our models?

One major issue with the simulations in this Section is that we have made some assumptions that might have biased the way our vocabulary networks

---

**Figure 11. The Structure of the Models in Program-2 and Program-3.**

```
Set up the network

[Apply a spike of activity]

Choose a word to activate

Turn the word ON

Update the network
```
behave. For example, we have chosen to activate words at random in the models in this section, but this has a number of important side-effects. One is that random activation will sometimes choose a word for activation that is already activated, and in that case the activation event will have no effect on the network. This is more likely to happen in a network where most of the words are already activated. For example, if 50% of the words in a network are already activated, then a randomly selected word has a 50% chance of being ON already, so half of our events will not make any substantial change to the activity level of the network. In a network where 90% of the words are already activated, only 1 event in 10 will produce an event that changes the network’s activity level. This suggests that networks with low levels of activity ought to be more likely to show change than networks where there is a higher level of activation. The effectiveness of an event should get smaller as the activation level in the network increases. Does this matter? It is hard to tell, but our model clearly has a built-in bias in this respect, and we need to be aware of that. Of course, we could re-code the model so that it selected only those words for activation that are currently inactive. Would this make a difference to the way the models work? It is hard to tell without actually writing the code and testing it out. Again, randomly selecting a word for activation means that all words have an equal chance of being activated by an event, but we could rewrite the code so that some words were more likely to be activated than others. For example, we could program the simulations so that the first 50 words in each network are more likely to be activated than the remaining words. Would this make a difference to the way the models work? Probably. A number of researchers, notably Carter (1987) and Brezina and Gablasova (2015), have suggested that vocabularies typically include a set of core words which are more resistant to attrition than other words are. We will come back to this assumption in a later section of the Workshop.

Another assumption we have made in this section is that we have chosen to work with models which have been allowed to settle into one of their stable attractor states before any events occur. However, it is far from obvious that a vocabulary network will normally be in its stable attractor state. Maybe we should assume that, by default, a vocabulary network is NOT one of its attractor states. Rather, the norm might be for vocabulary networks to be in unstable, excited state, where lots of words are flipping between the ON and the OFF state? Maybe this is what makes them work? Maybe the whole idea that an ideal vocabulary is one where most of the words are permanently ON is wrong? Maybe a better model would involve building a vocabulary network which had a naturally low-level of activity but responded very strongly to any input by activating large swathes of words very quickly? Figure 9 shows a network that seems to be able to switch quickly between a state of low activity when it experiences a lot of external stimulation, and maybe this is an important feature of real vocabularies?

Finally, we need to ask whether simply activating a word is a good way of modeling what happens when teachers provide exercises aimed at consolidating vocabulary. In the models in this section, a word that has been activated is treated just like any other word in that it will revert back to its “natural” state immediately if the activation state of other words does not allow it to remain activated. But we could argue that newly activated words are more resilient than this. Let us say that a newly activated word stays active for a few updates before it reverts to
its normal activity level. That would mean that an event that activates five words would effectively activate them for the next few updates as well. The effect of this would be enhanced when two or three events occur in close proximity to each other, and that would mean that the timing of events becomes a critical feature of the way these models work.

The important thing here is that working with the detail of the models has thrown up a number of considerations that were not apparent when we began, and we have ended up with a lot more questions than we started out with. This is typically what happens when you work with formal simulations. Every decision that you make in the programming has knock on effects on the way the simulations behave.

The obvious conclusion is that you need to be aware of the shortcomings of the simulation programs. Furthermore, you need to be prepared to develop new simulations that explore the implications of these shortcomings. It is worth pointing out that this is a very different way of doing research. Traditional “real-world” research is mainly difficult because it is hard to assemble large groups of participants and test them effectively. In simulation research, you do not need hundreds of participants: you just run the simulations many times with different parameter values. In real-world research, if your work throws up something that looks intriguing or suggestive, it is very unlikely that you will get the chance to repeat your study and collect the data you need to follow it up. With simulation research, repeat studies are very easy to do. This is important, because it often allows you to eliminate some unproductive lines of thought without going to the trouble of running actual experimental studies. The simulations in this section, for example, strongly suggest that a study that teaches students a small number of words in a one-off treatment is unlikely to produce lasting effects on their productive vocabulary knowledge. Looking at how often studies of this kind crop up in the current research literature should give you pause for thought.

6 Conclusion

We hope that this preliminary excursion into simulations and modeling will have shown you that this approach has a lot to offer to vocabulary researchers. Even a very simple model raises lots of questions, which would not often arise in the context of a more traditional research approach. In this section we have focused on the idea that increasing an L2 learner’s active vocabulary is a “good thing,” and we have tried to simulate one obvious approach to achieve this aim. The simulations have suggested that our traditional way of thinking about this might not be taking into account some fundamental properties of networks, and that vocabulary networks might be more resistant to change than what these traditional approaches suggest. We have also begun to question what is really going on when students are asked to carry out traditional vocabulary exercises. More importantly, perhaps, we find ourselves being pushed in the direction of seeing a vocabulary network as a dynamic structure, one where a lot of interesting things are taking place at the edges of the network. We will take these ideas further in the next Part of the Workshop.
References


