Making long-distance relationships work: Quantifying lexical competition with Hidden Markov Models

Julia Strand, David Liben-Nowell
Carleton College, United States

Abstract
A listener recognizes a stimulus word from acoustic–phonetic input by discriminating that word's representation from those of other words. The Neighborhood Activation Model (NAM; Luce & Pisoni, 1998) is a long-standing and deeply influential model quantifying how properties of the stimulus word and its competitors influence recognition. The current project incorporates Hidden Markov Models (HMMs) into the NAM's framework to more flexibly evaluate the influence of multiple lexical properties, thereby allowing us to pose novel questions about the process of spoken-word recognition. Analyses using HMMs' power to evaluate the stimulus's “distance” even to very distant words suggest that faraway words still act as competitors, suggesting that a larger subset of the lexicon is activated during recognition than has been previously assumed. Another analysis reveals that the way competition is distributed among other words significantly influences word recognition. HMMs have been widely applied in other domains, and our results demonstrate that they may be similarly suited to quantifying the processes underlying spoken-word recognition.

Introduction
Recognizing spoken words is an impressive perceptual and cognitive accomplishment; humans can process 250 words per minute (Foulke, 1968) and distinguish between timing differences in speech stimuli as short as 20 ms (Eimas & Corbit, 1973). In the last 60 years, numerous models have been developed to quantify the processes underlying spoken-word recognition (Luce, Goldinger, Auer, & Vitevitch, 2000; Luce & Pisoni, 1998; Marslen-Wilson, 1987; McClelland & Elman, 1986). The models are generally in agreement about several features of the word-recognition process. First, incoming acoustical input activates multiple lexical representations in memory. For example, when a listener hears a stimulus word “bird,” perceptually similar lexical entries (often called competitors or neighbors) such as “burn,” “heard,” and “bud” are also partially activated. In addition, the degree of activation is relative to the perceptual similarity between the stimulus word and the competitor (Luce & Pisoni, 1998; McClelland & Elman, 1986). When hearing “bird,” the highly confusable competitor “burn” is assumed to be more highly activated than the less confusable “bin.” The activation of the competitors influences recognition of the stimulus word, with greater activation from competitors hindering recognition. Therefore, words that are more perceptually distinctive, and therefore have less lexical competition, are recognized more quickly and accurately than those with more competition (Luce & Pisoni, 1998; Vitevitch & Luce, 1998).

In addition to the dynamics of lexical activation and competition, current models of spoken-word recognition also include mechanisms to account for the well-established effects of frequency of occurrence. Words that
quickly and accurately than those that are rare (Savin, 1963). Word frequency also appears to modulate lexical competition effects; words with low-frequency competitors are identified more quickly and accurately than those with high-frequency competitors (Luce & Pisoni, 1998). Most models assume that frequency acts by weighting activation levels of words prior to recognition or by biasing responses toward more frequent words following identification (see Dahan, Magnuson, & Tanenhaus, 2001).

To simultaneously represent the influence of frequency effects and lexical competition on spoken-word recognition, Luce and Pisoni (1986, 1998) proposed the Neighborhood Activation Model (NAM). The NAM has since become the most influential mathematical model of spoken-word recognition (see Magnuson, Mirman, & Harris, 2012 for a detailed discussion of the difference between mathematical models that seek to quantify lexical competition and other forms of computational models that simulate the process). Implementations of the NAM can readily generate predictions about how difficult specific words or classes of words will be to recognize, and these predictions may then easily be tested against human accuracy in spoken-word recognition. The flexibility of the NAM framework makes it an attractive candidate for testing novel predictions about the processes underlying spoken-word recognition. For example, the NAM predicts word recognition in cochlear implant users (Kirk, Pisoni, & Osberger, 1995), older adults (Sommers, 1996), and in visually perceived (lipread) speech (Auer, 2002) when the input to the model is changed to make it appropriate for the population or modality. Below, we outline the architecture of the NAM, identify issues in spoken-word recognition that the NAM has difficulty handling in its current form, and propose a novel method for quantifying the processes of lexical activation, and, therefore, competition.

Architecture of the Neighborhood Activation Model

The NAM assumes that word recognition involves discriminating among lexical representations that are activated in parallel. This process occurs using a collection of word decision units that simultaneously monitor three sources of information: the acoustic–phonetic input (bottom-up support for the lexical candidate), higher-level lexical information (word frequency), and the overall level of activity of other word decision units (lexical competition). Word recognition occurs when a specific word decision unit reaches a criterion and is discriminated from other activated lexical representations.

Given the high-level nature of the description that we have just given, there are many different operationalizations that are consistent with this model. As an anonymous reviewer of a previous version of this paper noted, it is important to distinguish the abstract verbal description of the process of word recognition from the numerical output of any particular formula motivated by that description. Some claims of the NAM, such as the beneficial effects of word frequency, are straightforward to turn into quantified predictions about word recognition. However, others, such as precisely what lexical competition means, are subtler. The original NAM paper quantifies competition in a particular way and those quantifications are tested in the paper. Here, we propose other methods of quantification that, while in keeping with the spirit of NAM (1998), make different predictions from the formulas contained therein. Comparing multiple ways of operationalizing lexical competition, each of which has implicit assumptions about the underlying process, allows us to pose questions about the process of spoken-word recognition.

As described by Luce and Pisoni (1998), the process within a word decision unit is quantified using Frequency-Weighted Neighborhood Probability (FWNP), which combines the influences of all three sources of information: acoustic–phonetic support, word frequency, and lexical competition. See Fig. 1 for the formula. Here \( \text{freq}(x) \) quantifies the frequency of occurrence of a word \( x \) (typically represented as the log of the number of occurrences per million words), and \( p(x|y) \) denotes the conditional probability of identifying a presented phoneme \( y \) as the phoneme \( x \) in a forced-choice phoneme identification task.

The numerator of FWNP\((w)\) is the Stimulus Word Probability (SWP), weighted by frequency. The SWP represents the bottom-up support for the stimulus word and can be thought of as a measure of intelligibility, because it quantifies the probability of perceiving the phonemes of the stimulus word given that those phonemes were presented. For example, the SWP of the word “bat” /bæt/ is \( \text{SWP}(\text{bat}) = p(\text{b}|\text{b}) \times p(\text{æ}|\text{æ}) \times p(\text{t}|\text{t}) \). The NAM posits that the decision units of words containing easily identified segments receive more support from the acoustic–phonetic input than words containing segments that are difficult to identify. For example, if participants correctly identify /w/ more often than /b/ on the forced-choice phoneme identification task, then the word “win” will have a higher SWP than the
word “thin,” and therefore, _ceteris paribus_, will be more likely to be correctly identified. Thus, the SWP reflects the likelihood of correctly identifying a stimulus word, based on the word’s segments themselves.

The denominator of FWNP(\(w\)) contains the summed frequency-weighted support for each lexical competitor \(w'\) (FWNP, Neighbor Word Probability) from the acoustic-phonetic input for \(w\), along with the frequency-weighted SWP (the support for the stimulus word itself). Competitors that are highly perceptually confusable with the stimulus word generate high NWPs, representing strong activation from the acoustic-phonetic input. Similarity is quantified in a similar manner to SWPs, again using the conditional probability of confusing the stimulus word's phonemes with the competitor's position-specific phonemes. For example, the probability of responding “meet” /mit/ given the stimulus “bead” /bid/ is calculated as \(\text{NWP(}\text{meet|}\text{bead)} = p(m|b) * p(i|i) * p(d|d).\)

Using this method, it is possible to calculate the NWP of any competitor word \(w'\) for a stimulus word \(w\) of the same length, even if the stimulus word and the competitor share no phonemes. But comparing words that differ in length (e.g., “meets” and “bead,” or “lease” and “least”) requires an additional step. To achieve this, _Luce and Pisoni (1998)_ included a “null response” category in the forced-choice consonant identification task. On some trials, no consonant was presented, and participants had “nothing presented” as a response category on all trials. Therefore, it is possible to determine the likelihood that participants would falsely report hearing a specific phoneme when nothing was presented (i.e., “hallucinating” that phoneme), or the likelihood that participants would respond that they had not heard anything when a specific phoneme was presented (i.e., missing the phoneme). Using the null-response data, _Luce and Pisoni (1998)_ are able to compute a quantity corresponding to the chances of perceiving, say, “meets” /mits/ when the actual stimulus presentation was “bead” /bid/: they align the vowels of the two words, and, working outwards from the vowel, evaluate the phoneme-by-phoneme similarities, using the null-response category where the words do not overlap. That is, they compute the probability of perceiving “meets” given “bead” as \(p(m|b) * p(i|i) * p(t|d) * p(s|\Omega).\) Conversely, the likelihood of perceiving “me” /mi/ given “bead” /bid/ — missing the /d/ — is computed as \(p(m|b) * p(i|i) * p(0|\Omega).\) In general, if we renumber the phonemes of a univocalic stimulus and competitor so that the vowel is the 0th phoneme of each (so that phonemes in the onset have negative indices, and those in the coda have positive indices), then we can express NWP as

\[
\text{NWP}(w'|w) := \prod_{i=-2}^{1} p(w'_{i}|w_{i})
\]

where \(p(w'_{i}|\Omega)\) and \(p(\Omega|w_{i})\), respectively, correspond to the probabilities of hallucinating \(w'_{i}\) and missing \(w_{i}\). (Note that, under this description, perceiving “scat” /skæt/ when presented with “cat” /kæt/ is the result of hallucinating an /s/ before the /k/.) An alternative description of this misperception is that we have mistaken the single phoneme /k/ for the diphone /sk/: in either case, the effect is to have perceived /skæt/ when the stimulus was /kæt/. The key difference between these ways of stating the process is that the /sk/-for-/k/ confusion implicitly includes the context in which the hallucination occurred: “the chance of hallucinating /s/ before a truly presented word-initial /k/” is identical to “the chance of hearing /sk/ when presented with a word-initial /k/.” But they may not be identical to “the chance of hallucinating /s/,” which omits the context. For more, see the paragraphs marked “Potential limitations in quantifying competition” in the Discussion.

To evaluate the total amount of lexical competition that a stimulus word encounters, _Luce and Pisoni (1998)_ calculated the NWPs of every English monosyllabic word, given any one of their (monosyllabic)\(^1\) stimulus words, and summed these values. Overall, then, the computed likelihood of correctly recognizing the stimulus word \(w\) is the value FWNP(\(w\), the proportion of the frequency-weighted support for words in the lexicon that the stimulus word contributes. The model predicts that recognition should be most difficult for words that have low SWP (that is, low predicted intelligibility), low frequency, and are perceptually similar to high-frequency competitors. Although the computations within the word decision units are relatively simple, the model has good predictive power; correlations between FWNP and word-recognition scores range from \(r = .23\) to \(r = .47\) (_Luce & Pisoni, 1998_).

By simultaneously combining the effects of predicted intelligibility, frequency, and lexical competition, the FWNPs makes specific, testable predictions about human word-recognition performance. For example, if a target word is preceded by a phonetically related prime, the residual activation from the prime should increase the NWP for that prime as a competitor, thereby reducing the target word FWNP. Indeed, priming a target word (e.g., “bull”) with the competitor with the highest NWP that does not share phonemes (e.g., “veer”) reduced identification accuracy (_Goldinger, Luce, & Pisoni, 1989_) and increased shadowing latencies (_Luce et al., 2000_) for the target. In addition, FWNPs can be predictive in situations that cause listeners to activate a particular set of words; FWNPs calculated for the names of drugs, using a lexicon of drug names rather than all English words, significantly predict clinicians’ identification accuracy in noise (_Lambert et al., 2005, 2010_). The mechanisms of activation and competition described by FWNPs are not limited to auditory speech; when the perceptual input is visual (lipread), FWNPs derive from visual confusion matrices predict lipread word-identification accuracy (_Auer, 2002; Feld & Sommers, 2011_).

Although FWNPs successfully predict human word-recognition performance, the simplicity of these computations makes some implicit assumptions about the quantification of lexical activation that may or may not have been intended by the creators of NAM. That is, some

---

\(^1\) The reference lexicon used by _Luce and Pisoni (1998)_ included only monosyllabic words. However, the authors specify that this restriction was imposed to simplify the computational analysis, rather than as a theoretical claim about which words are simultaneously activated. Although the calculations for the FWNP apply directly to polysyllabic univocalic words (“cattle” as a competitor for “cat”), as implemented by _Luce and Pisoni (1998)_), FWNP cannot be computed for words with multiple vowels without modification.
components of the FWNPs may reflect computational challenges rather than theoretical claims. As one example, the NWP for “cast” /kæst/ given the stimulus word “cat” /kæt/ is 
\[ p(k|k) + p(æ|æ) + p(s|t) + p(t|0) \], representing the participant correctly hearing /k/ and /æ/, then mistaking /s/ for /t/, and finally hallucinating a /t/ when nothing was presented. Although this sequence certainly is one way that the two words could have been mistaken, it is also possible that the error was a single hallucination of an /s/ (while perceiving every other phoneme correctly), quantified as 
\[ p(k|k) + p(æ|æ) + p(s|0) + p(t|t) \]. The overall confusability of word pairs (and therefore degree of lexical competition for a particular stimulus word) may be more accurately represented by a metric that considers other possible ways of confusing the words.

The NAM successfully predicts many phenomena in human word recognition and has been very influential in the field. Incorporating more flexible methods for computing perceptual similarity and lexical competition will allow us to extend the NAM framework to ask additional questions about the processes underlying spoken-word recognition. Indeed, Magnuson et al. (2012, p. 79) suggest that “using other similarity metrics in the NAM framework would be an excellent strategy for making further progress on identifying general constraints on spoken-word recognition.”

**A more flexible framework for measuring competition: Hidden Markov Models**

The NAM framework allows us to focus our attention on one concrete computational task: given a stimulus word \( w \) and any particular competitor word \( w' \), we must compute the probability that a listener would hear \( w' \) instead of \( w \). There are two distinct difficulties for doing this computation in the basic NAM framework. First, we seem to get the wrong probabilities for some words: the NWP assumes a particular structure in the perceptions and misperceptions that caused \( w' \) to be heard instead of \( w \). Namely, hallucinated and missing phonemes can occur only at the periphery of the word (as in the “cast|cat” example above), while the phonemes adjacent to the aligned vowel may only be mistaken for each other. That assumed structure may not be the most probable way of confusing \( w \) and \( w' \). Second, we cannot even perform this computation for every potential competitor: for those competitors that contain multiple vowels, the NWP does not allow us to compute a confusion probability at all because “aligning the vowels of \( w \) and \( w' \)” is not well defined.

Here, we suggest a generalization in the NAM framework to simultaneously address these issues by adapting our setting to a probabilistic modeling approach called Hidden Markov Models (HMMs; see Jurafsky & Martin, 2008; Russell & Norvig, 2009 for an introduction to this technique). HMMs have been used successfully in a wide range of domains, including computational biology (Eddy, 2004; Yoon, 2009), visual perception (Brand, Oliver, & Pentland, 1997), and automatic speech recognition (Gales & Young, 2007; Rabiner, 1989; see also Scharenborg, 2007 for links between automatic and human speech recognition), but have not yet been applied to quantifying lexical competition. Intuitively, a Hidden Markov Model is an abstract mathematical “machine” that describes a probabilistic process that generates an “output” (in our setting, a sequence of phonemes). We will construct an HMM for every word in the lexicon, with the goal that the machine for \( w \) will generate \( w' \) as output with higher probability the more perceptually similar \( w \) and \( w' \) are.

To be concrete, we will describe the HMM for a particular word, “cat” /kæt/. The HMM \( M_{cat} \) consists of a collection of states: one state for each of the three phonemes in “cat,” and four “hallucination states” before and after each of these phonemes. We also add a start state and an end state (see Fig. 2).

Each of these states is associated with two probability distributions: the emission probabilities (for example, when the machine is in the /æ/ state, what is the probability that the phoneme /æ/ will be generated? Or /o/? Or no phoneme at all?) and transition probabilities (if the machine is currently in the /æ/ state, what is the probability that it will be in the /t/ state in the next step? Or hallucination state 3?). Our probability distributions are derived experimentally, as we discuss below. “Running” an HMM \( M_{cat} \) means visiting a sequence of states of the machine, generating a phoneme (or Ø) in each state that we visit. More concretely, we begin in the start state, and repeatedly (a) append to the output sequence a phoneme chosen according to the current state’s emission probabilities; and (b) move to a new state, chosen according to the current state’s transition probabilities. We continue to run the machine until it arrives in the final state. This process results in the machine generating a particular output sequence of phonemes.

Note that the HMM \( M_{cat} \) can generate many different outputs, depending on the probabilistic choices that happen to be made during the run. For example, \( M_{cat} \) can generate “cat” by following the state sequence start \( \rightarrow /k/ \rightarrow /æ/ \rightarrow /t/ \rightarrow \text{end} \), and successively generating Ø, /k/, /æ/, /t/, Ø in those five states. Or \( M_{cat} \) can generate “claps” by following the state sequence start \( \rightarrow /k/ \rightarrow 2 \rightarrow /æ/ \rightarrow /t/ \rightarrow 4 \rightarrow \text{end} \), and successively generating Ø, /k/, /l/, /æ/, /p/, /s/, Ø in those seven states. (Note that in the latter run the state /t/ generated /p/: a less probable phoneme for the /t/ state to generate than /t/, but a possibility.) A crucial mathematical fact about an HMM is that

---

2 Given the data we have, the emission probability for phoneme \( x \) in a particular state \( y \) is the confusion probability \( p(y|x) \). Phonemic context could be incorporated into an HMM by modifying the confusion probabilities for each particular state based on one or more preceding phonemes. To add one phoneme of context, for example, the probability of hallucinating /s/ in State 3 in Fig. 2 (to generate the word “cast”) would be set to the (experimentally derived) probability of hallucinating an /s/ immediately after /æ/; the probability of hallucinating an /s/ in State 4 (to generate the word “cats”) would be the (experimentally derived) probability of hallucinating an /s/ immediately after /æ/. Similarly, the emission probabilities in the /æ/ state of \( M_{cat} \) would differ from the emission probabilities in the corresponding /æ/ state of \( M_{as} \).

3 We can observe the phonemes in the output, but which state the HMM is in at any particular time is unobservable; the “hidden” in the name “Hidden Markov Model” derives from this fact.
\[
\sum_{\text{phonemic sequences } w'} p(w' | M_w) = 1
\]

i.e., that the quantities \( p(w' | M_w) \), the probabilities of the machine generating any particular output sequence \( w' \), in fact form a probability function. This fact allows us to reason about that probability distribution rigorously. Intuitively, a competitor \( w' \) that is generated with a higher probability by the HMM \( M_w \) is more similar to \( w \).

Importantly, an HMM can generate the same phonemic sequence in multiple ways — that is, via multiple paths (sequences of hidden states) through the machine. For example, the word “cast” could be generated along a more probable path, following the state sequence \( \text{start} \rightarrow [k] \rightarrow [æ] \rightarrow [t] \rightarrow 4 \rightarrow \text{end} \), where state \( [t] \) generated \( /s/ \) and state 4 (the last hallucination state) generated \( [t] \). That path corresponds to what the original NAM computes. Or “cast” could be generated along a much less probable path, \( \text{start} \rightarrow [k] \rightarrow [æ] \rightarrow [t] \rightarrow 4 \rightarrow 4 \rightarrow \text{end} \), where state \( [k] \) generated \( /æ/ \), state \( [æ] \) generated \( /æ/ \), state \( [t] \) generated \( [k] \), and state 4 successively generated \( /æ/ \), \( /s/ \), and \( /t/ \). The probabilities of these respective sequences are

\[
p(\text{non-hall}) \cdot p([k]) \cdot p(\text{non-hall}) \cdot p(\text{non-hall}) \cdot p(\text{non-hall}) \cdot p(\text{non-hall}) \cdot p(t) \cdot p(\text{hall}) \cdot p(\text{hall}) \cdot p(s) \cdot p(\text{hall}) \cdot p(t) \cdot p(\text{hall})
\]

where \( p(\text{hall}) \) and \( p(\text{non-hall}) \) denote the probability of hallucinating and not hallucinating, respectively. Unlike the original NAM formulation, the HMM imposes no assumption on the “alignment” of a competitor word to the baseline stimulus word from which the machine was constructed. Instead, we can simultaneously consider all possible ways that a stimulus word \( w \) can be mistaken for a competitor \( w' \). Using versions of two well-known algorithms for HMMs, called the Viterbi Algorithm and the Forward Algorithm (see Jurafsky & Martin, 2008; Russell & Norvig, 2009), we can efficiently compute, for any particular HMM \( M_w \) and any particular word \( w' \), two key quantities: (a) the maximum path probability of \( w' \) (out of all paths through \( M_w \) that generate \( w' \), what is the probability of the most probable path?); and (b) the total probability of \( w' \) (out of all paths through \( M_w \) that generate \( w' \), what is the sum of the probabilities of these paths?). We can use both of these quantities as measures of the confusability of words \( w \) and \( w' \).

**The current study**

A key principle of the NAM is that word identification is influenced by “the number and nature of lexical items activated by the stimulus input” (Luce & Pisoni, 1998, p. 12). Using HMMs in the NAM framework will allow us to further explore the processes underlying spoken-word recognition by more flexibly evaluating the number and nature of activated representations. Below, we describe four novel research questions that may be addressed using HMMs.

**Aim 1. Measuring perceptual intelligibility**

The NAM and other models of word recognition posit that lexical activation is a function of the perceptual similarity between the acoustic–phonetic input and the lexical item in memory. In the NAM, intelligibility is calculated as

\[
\sum_{\text{perceptual support for the stimulus word from the stimulus input, e.g., the SWP of “bat”/bæt/ is } p(b|b) \ast p(æ|æ) \ast p(t|t). \text{ However, it is possible for a word to be identified correctly despite misperceptions (for “right for the wrong reasons”): e.g., to identify “bat” as “bat” but doing so by both missing the /b/ and then immediately after hallucinating a /b/. HMMs enable us to evaluate multiple paths to correct recognition. This capability will enable us to evaluate whether having other paths to “correct” recognition facilitates recognition because there are more ways to hear the right word, or hinders recognition because other paths represent a kind of “self-confusability.”}
\]
**Aim 2. Measuring perceptual confusability**

Quantifying perceptual similarity effectively is a critical component to modeling lexical competition. To the best of our knowledge, no studies to date have included multiple routes for confusing pairs of words, as in the example above. This omission is likely due to computational difficulty, rather than theoretical motivation. Therefore, we will use HMMs to more flexibly represent the perceptual similarity of word pairs, computing all of the ways in which a pair of words can be confused for each other rather than just one.

**Aim 3. Quantifying the spread of lexical activation**

This project also aims to make progress in evaluating the extent to which activation spreads throughout the lexicon. Although models of word recognition agree that highly perceptually similar competitors receive the most activation, it is unclear whether even relatively dissimilar words are also activated to some degree. In other words, the extent to which lexical activation diffuses throughout the lexicon remains unknown. In the original presentation of NAM, all stimulus words were consonant–vowel–consonant (CVC) words, and the reference lexicon included all monosyllabic words. Therefore, for the stimulus word “beat” /bit/, “be” /bi/ could act as a competitor but “beetle” /bidl/, a disyllabic word containing a syllabic /l/, could not. Simplified implementations of NAM have quantified lexical competition using the Deletion–Addition–Subtraction (DAS) rule, considering only words that differ by a single phoneme from the stimulus word: competitors of “beat” /bit/ include “be” /bi/ and “bleat” /blit/ and “meat” /mit/ but not “need” /nid/.

Given that HMMs allow distance computation between any two words, we can consider any subset of the lexicon as possible competitors for the stimulus word. To evaluate the extent to which activation spreads through the lexicon, we will include as the set of competitors all monosyllabic words. Therefore, for the stimulus word “beat” /bit/, “be” /bi/ could act as a competitor but “beetle” /bidl/, a disyllabic word containing a syllabic /l/, could not. Simplified implementations of NAM have quantified lexical competition using the Deletion–Addition–Subtraction (DAS) rule, considering only words that differ by a single phoneme from the stimulus word: competitors of “beat” /bit/ include “be” /bi/ and “bleat” /blit/ and “meat” /mit/ but not “need” /nid/.

**Aim 4. Evaluating the source of lexical competition**

Recall that FWNP(w) is the ratio between the (frequency-weighted) support for w and the (frequency-weighted) support for all words, including w and all competitors w. The competition component of the FWNP’s denominator simply sums the competition from all competitors. In doing so, it loses information about whether the majority of the competition came from a few highly similar competitors or a larger number of less similar competitors. For example, “bone” and “doom” have reasonably similar frequencies of occurrence, SWPs, and FWNP values, but they differ in the source of the competition (see Fig. 4).

Strand (2014) demonstrated that the dispersion of competition had a small but significant effect on word-recognizability after controlling for the amount of overall lexical competition; stimulus words like “bone” whose competitors are more tightly clustered around the mean level of the stimulus’s competition were recognized more accurately than words like “doom” that have more high-competition competitors (and thus more low-competition competitors too). In the current study, we focus on the extremes of these distributions of competition by evaluating the influence on recognition of the proportion of the total competition a word encounters that comes from its closest competitor.

**Methods**

The NAM and HMM calculations require measures of phoneme confusability as input. To test the predictions of the models, we also require measures of word-recognizability accuracy.

**Phoneme-identification**

Phoneme-identification data were obtained from an existing dataset (Luce, 1986; Luce & Pisoni, 1998). The stimuli consisted of 25 consonants (b, tʃ, d, f, g, h, dʒ, k, l, m, n, ɲ, p, r, s, ʃ, t, θ, ð, v, w, j, z, ʒ) and 15 vowels (i, ɪ,
The consonant task also included a “null” condition (Luce & Pisoni, 1998, described above) in which a vowel was presented in isolation but participants were given the opportunity to report that a consonant had been presented in addition to the vowel. The null condition renders values for perceptual hallucinations and omissions that enable us to quantify the similarity of words with differing lengths. The vowel data set included the rates at which participants failed to report the vowel they heard (representing omissions), but did not include a null stimulus category, so it is not possible to estimate the likelihood of vowel hallucinations. Thus, vowel-hallucination probabilities are set to zero in the present work. (For the implications of this zero probability, see the paragraphs marked “Potential limitations in quantifying competition” in the Discussion.)

Luce and Pisoni (1998) presented participants with phonemes to identify at three different signal-to-noise (SNR) ratios, \(-5, 5, \) and \(15\). Prior work has shown that although SNR affects the accuracy with which phonemes are identified, it does not systematically change the types of confusions that are made (see Miller & Nicely, 1955 for evidence that consonant confusions are consistent across SNRs). Therefore, in the current study, confusion matrices were collapsed across the three matrices to increase the number of data points in each cell and reduce the effects of small idiosyncrasies in an individual confusion matrix. Luce and Pisoni (1998) also differentiated between consonant-initial and consonant-final confusions. However, this classification becomes problematic when comparing words of longer lengths; for example, should the “r” of “carry” be treated as a final consonant (because it follows “a”) or an initial consonant (because it precedes “y”)? In the current
study, we collapsed across these two categories to create a single context-independent quantification of the confusability of consonant pairs.

Word recognition

Participants

53 native English speakers with self-reported normal hearing and normal or corrected-to-normal vision were recruited from the Carleton College community. Carleton College's Institutional Review Board approved the research procedures.

Stimuli & procedures

Stimuli included 400 CVC words selected from the English Lexicon Project (ELP; Balota et al., 2007). The talker was a Midwestern female, and recording was done at 16 bit, 44,100 Hz using a Shure KSM-32 microphone with a pop filter. Stimuli were equated for RMS amplitude using Adobe Audition, version 5.0.2, and presented in isolation through Seinheisser HD-280 headphones in background noise (six-talker babble), set at 65 dB SP at SNR of 0. Participants were seated in a quiet room at a comfortable distance from a 21.5” monitor. Stimulus presentation was controlled with Superlab, version 5. Participants identified
stimuli by typing their responses on a keyboard. They were encouraged to guess when unsure. Prior to analysis, recognition responses were hand-checked for obvious entry errors, such as a superfluous punctuation mark (e.g., “soup[“]). Entry corrections accounted for approximately 1% of responses. No other deviations from the stimulus word (plurals, inflected forms) were counted as correct. This dataset has been reported on previously by Slote and Strand (2015).

Quantifying lexical competition

A lexicon of potential competitors was constructed from a list of 40,411 English words from the ELP (Balota et al., 2007), with two modifications. First, we replaced each /a/ by /æ/ to make pronunciations consistent with the phoneme-identification data (Luce & Pisoni, 1998). Frequency counts were derived from the Subtlex norms (Brysbaert & New, 2009). Second, we combined entries for homophones, where the combined entry’s frequency (Brysbaert & New, 2009) resulted in 19,860 observations. An equivalent statistic to NN(w) := argmaxw≠w[p_all(w’|w) * freq(w’)]

i.e., NN(w) is the competitor w’ of w that is most likely to be generated by the HMM M_w, weighted by frequency. Using these raw probabilities, we compute several measures for each stimulus word w (see Table 1).

We also consider restrictions to the set of competitors for a stimulus word w, giving us three variations of the conf_all measure (see Table 2).

Finally, we also compute two measures using the original NAM model of distance (SWP and NWP), instead of the HMM-based p_all (see Table 3). The code (written in Python) used to generate all of these measures, along with values for each variable described here for all English CVCs, are available at <http://go.carleton.edu/StrandLab>.

Results

Mixed-effect models with a binomial distribution were created in R (R Core Team, 2013) and the R packages lme4 and languageR (see Baayen, 2008) were used to evaluate the influence of multiple lexical predictors on the criterion variable, word-recognition accuracy (“correct” vs “incorrect”). For each lexical variable, by-participant random slopes were included when they improved the fit of the model. Variables were centered around their means, and likelihood ratio tests (Baayen, Davidson, & Bates, 2008) were used to evaluate whether models of increasing complexity provided better fits for the data. Excluding words that were skipped by participants and those that were not presented due to experimental error (6% of trials) resulted in 19,860 observations. An equivalent statistic to R^2 does not currently exist for evaluating the fit of a logistic model, and values derived by the multiple measures of proposed pseudo-R^2 can vary significantly (Long & Freese, 2001). Thus, assessing model fit for these type of data is notoriously difficult (Gelman & Hill, 2007). To evaluate the relative improvements caused by adding new variables to the model, we report the reduction in Akaike’s Information Criterion (AIC). AIC reductions caused by novel vari-

### Table 1

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mathematical definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>intel_all</td>
<td>p_all(w</td>
<td>M_w)</td>
</tr>
<tr>
<td>intel_max</td>
<td>p_max(w</td>
<td>M_w)</td>
</tr>
<tr>
<td>conf_all</td>
<td>( \sum_{w \neq w'} p_{all}(w'</td>
<td>M_w) \cdot \text{freq}(w') )</td>
</tr>
<tr>
<td>NN_ratio</td>
<td>( \frac{\text{freq}(\text{NN}(w)/\text{conf_all})}{\text{conf_all}} )</td>
<td>Fraction of (log frequency-weighted) confusability coming from the largest term (that is, the competitor w with the largest value of ( p_{all}(w'</td>
</tr>
</tbody>
</table>

### Table 2

Variations on the conf_all measure, based on the subset of the lexicon included.

<table>
<thead>
<tr>
<th></th>
<th>Mathematical definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>conf_all</td>
<td>( S ) = entire lexicon (except w itself)</td>
</tr>
<tr>
<td>conf_mono</td>
<td>( S ) = all monosyllabic words (except w itself)</td>
</tr>
<tr>
<td>conf_DAS</td>
<td>( S ) = all words that are one deletion/addition/substitution away from w</td>
</tr>
<tr>
<td>conf_CV</td>
<td>( S ) = all words with the same consonant/vowel pattern as w (except w itself)</td>
</tr>
</tbody>
</table>

### Table 3

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mathematical definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>intel_NAM</td>
<td>SWP(w) := ( \prod_{i=1}^{n} p(w_i</td>
</tr>
<tr>
<td>conf_NAM</td>
<td>( \sum_{w \neq w'} [\text{NWP}(w'</td>
</tr>
</tbody>
</table>

where NWP(w|w) := \( \prod_{i=1}^{n} p(w_i'|w) \) and S = all monosyllabic words (except w)
ables can be contextualized by comparing them to AIC reductions from well-established variables, such as word frequency.

**Baseline model**

A model that contained only subjects and items as random effects was first constructed to serve as a comparison point for other models. As would be expected, adding centered, log-transformed frequency values (Brysbaert & New, 2009) as a fixed factor significantly improved the fit of the model, $\chi^2(1) = 32.97$, $p < .001$, AIC reduction = 30.9; higher word frequency was associated with greater likelihood of word-recognition accuracy.

Given that frequency was entered as a control predictor, by-subject random slopes for frequency were not included (see Barr, Levy, Scheepers, & Tily, 2013 for more on this issue). See Table 4.

The multiple measures of predicted intelligibility and lexical competition are derived from the same confusion matrix with small modifications, resulting in high correlations among some of the measures. When correlated variables are simultaneously entered in the same model, the high degree of collinearity can complicate evaluating the unique contribution of each measure (Friedman & Wall, 2005). Therefore, in this case, a model that includes fixed factors X1 & X2 combined is compared to a model that includes only fixed factor X1, and one that includes only fixed factor X2. Given the collinearity between the measures, the coefficient estimates of such a model are not interpretable, but the likelihood ratio tests can reveal whether including multiple fixed factors explains additional variance beyond the single factors. The analyses below evaluate the influence of multiple lexical variables, and are compared to the frequency-only model.

**Analysis 1: Measuring perceptual intelligibility**

Our first aim was to evaluate the predictive power of multiple measures of word intelligibility. These included intel_NAM, intel_all, and intel_max centered around their means. First, the effect of each measure was evaluated individually, by adding it as a fixed factor to the baseline model that included frequency as a fixed effect and subjects and items as random effects.

When entered individually, all measures of predicted intelligibility improved the fit of frequency-only model, intel_NAM: $\chi^2(1) = 61.88$, $p < .001$, AIC reduction = 55.9; intel_all: $\chi^2(1) = 61.34$, $p < .001$, AIC reduction = 55.9; and intel_max: $\chi^2(1) = 61.88$, $p < .001$, AIC reduction = 55.9. Including by-subject random slopes for each measure of intelligibility improved the fit of the models, indicating that participants differ in the extent to which they are influenced by effects of lexical intelligibility, intel_NAM: $\chi^2(1) = 11.6$, $p < .001$, AIC reduction = 7.6; intel_all: $\chi^2(1) = 11.57$, $p < .001$, AIC reduction = 7.6; intel_max: $\chi^2(1) = 11.6$, $p < .001$, AIC reduction = 7.6. Note that the magnitude of the AIC reductions caused by adding intelligibility measures were numerically larger than the well-established effect of frequency, indicating strong effects of word intelligibility. Intel_NAM and intel_max are perfectly correlated: for stimuli of the same length, they only differ by a linear transformation (the probability of failing to hallucinate phonemes anywhere in the word) to make the latter true probabilities. Therefore, the results are identical when intel_NAM is substituted for intel_max. Although intel_NAM and intel_max are perfectly correlated for CVC-only stimuli, a benefit of intel_max is that it yields scores “on the same scale” as the HMM-based confusion measures that can in principle be applied to polyvocalic words (where the align-the-vowels step of NAM is not well-defined). Therefore, intel_max is included in subsequent models.

To evaluate whether intel_max and intel_all contribute uniquely to the model, both predictors were simultaneously added to the frequency-only model as fixed factors. A model that included both intel_max and intel_all provided a better fit than one that included only intel_max: $\chi^2(1) = 7.96$, $p = .005$, AIC reduction = 6.0; or only intel_all: $\chi^2(1) = 8.45$, $p = .004$, AIC reduction = 6.5, indicating that intel_max and intel_all are both contributing uniquely to the model, although the size of the effect is somewhat modest compared to the more robust effect of frequency.

By-subject random slopes for both models of predicted intelligibility improved the fit to the data, intel_all: $\chi^2(1) = 11.54$, $p < .003$, AIC reduction = 7.6; intel_max: $\chi^2(1) = 11.58$, $p = .003$, AIC reduction = 7.6, but a model with by-subject random slopes for both failed to converge.

Given the high degree of collinearity between intel_max and intel_all, the estimates of the coefficients are not interpretable, so it is not clear whether the two predictors are affecting recognition in the same direction. That is, are higher values for intel_all and intel_max associated with higher or lower rates of word identification? To render the directions of the effects interpretable, we residualized intel_all by conducting a simple linear regression, predicting intel_all from intel_max, to generate intel_all_resid. This step will enable us to enter intel_max into the model, along with intel_all_resid, giving intel_max (the variable that has been tested previously in the literature, in the form of intel_NAM) first access to the shared variance. Although there are circumstances under which orthogonalizing predictor variables by residualizing one variable against another is not appropriate (Wurm & Fisicaro, 2014), the benefit of this approach is that it allows us to simultaneously evaluate the unique contribution of intel_all on explaining word recognition scores, beyond the variance explained by intel_max. Importantly, this analysis will generate the same results for the predictor that was not residualized (intel_max) as being entered in the model alone. In addition, the coefficient estimate for the

<table>
<thead>
<tr>
<th>Fixed effects</th>
<th>Estimate</th>
<th>SE</th>
<th>z value</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>0.24</td>
<td>0.09</td>
<td>2.65</td>
<td>0.008</td>
</tr>
<tr>
<td>frequency</td>
<td>0.45</td>
<td>0.08</td>
<td>5.85</td>
<td>&lt;0.001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>deviance</th>
<th>df.resid</th>
</tr>
</thead>
<tbody>
<tr>
<td>22253.1</td>
<td>22284.7</td>
<td>-11122.5</td>
<td>22245.1</td>
<td>20,174</td>
</tr>
</tbody>
</table>

Table 4: Summary of baseline model with frequency.
residualized predictor (intel_allresid) alone will be the same as when it is simultaneously included with the non-residualized predictor (intel_max). The residualization will not improve the overall explanatory power of the model nor any indices of model fit, but will enable interpretation of the coefficient estimates. For more detail on the consequences of residualization, see Wurm and Fisicaro (2014). The output of this model is shown in Table 5.

The sign of the estimate of intel_max is positive, indicating that higher intelligibility values are associated with higher accuracy, as expected. However, intel_allresid is negative, indicating that higher values are associated with lower accuracy. These results suggest that the most intelligible words have high probabilities along the highest probability path, and low probability along other paths to correct identification (but via incorrect paths).

Analysis 2: Measuring perceptual confusability

Our second question was whether evaluating lexical competition between a given stimulus word and a competitor in multiple ways (e.g., the “cast|cat” example described above) would improve the predictive power of the model over the lining-up-the-vowel method. To evaluate this question, we compared the effects of conf_mono and conf_NAM. Both of these variables include the same subset of the lexicon (all monosyllabic words) but they differ in that conf_mono allows confusions between stimulus word and competitor in multiple ways. Models that contain only conf_mono or conf_NAM both provided a better fit than the frequency-only model, conf_mono: $\chi^2(1) = 51.36, p < .001$, AIC reduction = 49.4; conf_NAM: $\chi^2(1) = 49.29, p < .001$, AIC reduction = 47.3. A model that included both conf_mono and conf_NAM did not provide a better fit than conf_mono alone, $\chi^2(1) = 0.82, p = .37$, AIC reduction = –1.2. The model with both conf_mono and conf_NAM performed slightly better than conf_NAM alone, but the difference was only marginally significant, $\chi^2(1) = 3.15, p = .08$, AIC reduction = 1.2. This result indicates that the method of quantifying lexical competition from the HMM accounts for only a marginal degree of unique variance in word-recognition accuracy beyond that explained by the original NAM method.

Analysis 3: Quantifying the spread of lexical activation

The third aim was to assess the extent to which lexical activation spreads through the lexicon. To evaluate this question, we calculated lexical competition using differing subsets of the lexicon as potential competitors: DAS neighbors only (conf_DAS), substitution-only neighbors (conf_CV), monosyllabic words only (conf_mono); and all words in the lexicon (conf_all). Each of the measures on its own added significant unique variance to the frequency-only model, conf_DAS: $\chi^2(1) = 12.89, p < .001$, AIC reduction = 10.9, conf_CV: $\chi^2(1) = 44.83, p < .001$, AIC reduction = 42.9, conf_mono: $\chi^2(1) = 51.36, p < .001$, AIC reduction = 49.4, conf_all: $\chi^2(1) = 52.30, p < .001$, AIC reduction = 50.3.

Next, we included multiple measures of competition simultaneously. A model that included both conf_CV and conf_DAS as fixed effects provided a better fit than one that included only conf_CV, $\chi^2(1) = 12.36, p < .001$, AIC reduction = 10.3 or a model that included only conf_DAS, $\chi^2(1) = 44.29, p < .001$, AIC reduction = 42.3, suggesting that words that are an addition or deletion away from the target are providing competition. A model that included conf_mono in addition to conf_CV and conf_DAS accounted for additional variance beyond the model with conf_CV and conf_DAS, $\chi^2(1) = 19.77, p < .001$, AIC reduction = 17.8. Finally, including the full lexicon (conf_all) in addition to conf_CV, conf_DAS, and conf_mono provided a better fit than the model without it, $\chi^2(1) = 15.97, p < .001$, AIC reduction = 14.0, suggesting that multisyllabic words outside of the DAS neighborhood are providing competition for the stimulus words.

In order to render interpretable estimates of the coefficients, we again residualized variables. Given that the most research has been done on the effect of DAS neighbors, we did simple linear regressions on conf_CV (removing the variance explained by conf_DAS, conf_mono, and conf_all), conf_mono (removing the variance explained by conf_DAS, conf_CV, and conf_all), and conf_all (removing the variance explained by conf_DAS, conf_CV, and conf_mono). Then, we entered conf_DAS into a model, along with conf_CVresid, conf_monoresid, and conf_allresid. This analysis will render an estimate of the coefficient for conf_DAS that is the same as conf_DAS being entered in alone (see Table 6).

All four measures have negative estimates, indicating that competition from all subsets of the lexicon is negatively correlated with word-recognition accuracy.

Analysis 4: Evaluating the source of lexical competition

The fourth aim was to better understand how the distribution of competition across competitors influences

### Table 5

Evaluating the contributions of intel_max and intel_allresid.

<table>
<thead>
<tr>
<th>Fixed effects</th>
<th>Estimate</th>
<th>SE</th>
<th>z value</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>0.24</td>
<td>0.09</td>
<td>2.76</td>
<td>.01</td>
</tr>
<tr>
<td>frequency</td>
<td>0.44</td>
<td>0.07</td>
<td>6.14</td>
<td>&lt;.001</td>
</tr>
<tr>
<td>intel_max</td>
<td>34.04</td>
<td>4.70</td>
<td>7.24</td>
<td>&lt;.001</td>
</tr>
<tr>
<td>intel_allresid</td>
<td>-0.20</td>
<td>0.07</td>
<td>-2.84</td>
<td>.005</td>
</tr>
</tbody>
</table>

### Table 6

Evaluating the contributions of conf_DAS, conf_CVresid, conf_monoresid, and conf_allresid.

<table>
<thead>
<tr>
<th>Fixed effects</th>
<th>Estimate</th>
<th>SE</th>
<th>z value</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>0.24</td>
<td>0.08</td>
<td>2.91</td>
<td>.004</td>
</tr>
<tr>
<td>frequency</td>
<td>0.54</td>
<td>0.07</td>
<td>7.84</td>
<td>&lt;.001</td>
</tr>
<tr>
<td>conf_DAS</td>
<td>-5.70</td>
<td>1.41</td>
<td>4.03</td>
<td>&lt;.001</td>
</tr>
<tr>
<td>conf_CVresid</td>
<td>-1.10</td>
<td>0.16</td>
<td>-7.07</td>
<td>&lt;.001</td>
</tr>
<tr>
<td>conf_monoresid</td>
<td>-14.55</td>
<td>1.75</td>
<td>-8.32</td>
<td>&lt;.001</td>
</tr>
<tr>
<td>conf_allresid</td>
<td>-14.70</td>
<td>1.74</td>
<td>-8.47</td>
<td>&lt;.001</td>
</tr>
</tbody>
</table>

### Table 7

Analysis 4: Evaluating the source of lexical competition.

<table>
<thead>
<tr>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>deviance</th>
<th>df(resid)</th>
</tr>
</thead>
<tbody>
<tr>
<td>22198.8</td>
<td>22246.3</td>
<td>-11093.4</td>
<td>22186.8</td>
<td>20,172</td>
</tr>
</tbody>
</table>
recognition. Do recognition rates differ between words whose competition comes primarily from a single, highly similar neighbor, versus coming from a larger number of less similar neighbors? To that end, we first built models that include only the total competition (conf_all) and the proportion of total competition that comes from the single competitor that provides the most competition (NN_ratio). Conf_all provided a better fit than the frequency-only model, $\chi^2(1) = 52.30, p < .001$, AIC reduction = 50.3; but NN_ratio did not, $\chi^2(1) = 2.73, p = .10$, AIC reduction = .08.

To evaluate the independent contributions of the proportion of competition from the nearest neighbor and the competition from the other neighbors, we tested a model that included both conf_all and NN_ratio. This model provided a better fit than conf_all alone, $\chi^2(1) = 6.59, p = .01$, AIC reduction = 4.6; or NN_ratio alone, $\chi^2(1) = 56.16, p < .001$, AIC reduction = 54.1. Because conf_all and NN_ratio are not correlated ($r = .08$), it was not necessary to generate a residualized measure to test the direction of the effects as in the previous analyses. Thus, we report the summary of the model with conf_all and NN_ratio entered simultaneously in Table 7.

These results indicate that the amount of competition that comes from the nearest neighbor accounts for a small but significant amount of unique variance in word-recognition accuracy, beyond that explained by the total amount of competition. After controlling for the total amount of competition, words with a highly similar neighbor (those with a larger fraction of their competition coming from a single competitor) are easier to recognize than those with less competition coming from the closest competitor.

**Table 7**

<table>
<thead>
<tr>
<th>Fixed effects</th>
<th>Estimate</th>
<th>SE</th>
<th>z value</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>0.25</td>
<td>0.09</td>
<td>2.85</td>
<td>.004</td>
</tr>
<tr>
<td>frequency</td>
<td>0.53</td>
<td>0.07</td>
<td>7.30</td>
<td>&lt;.001</td>
</tr>
<tr>
<td>conf_all</td>
<td>-8.44</td>
<td>1.09</td>
<td>-7.76</td>
<td>&lt;.001</td>
</tr>
<tr>
<td>NN_ratio</td>
<td>2.59</td>
<td>1.00</td>
<td>2.58</td>
<td>.01</td>
</tr>
<tr>
<td>AIC</td>
<td>22198.2</td>
<td></td>
<td>11093.1</td>
<td></td>
</tr>
<tr>
<td>BIC</td>
<td>22245.6</td>
<td></td>
<td>22186.2</td>
<td>20,172</td>
</tr>
</tbody>
</table>

Measuring perceptual intelligibility

First, measures of perceptual similarity of word pairs that include multiple possibilities for confusability are richer predictors than those that compute similarity in only one way. Our results indicate that words that have a single clear path to recognition are recognized most easily, and that having multiple routes to correct recognition actually hinders the listener. This metric of “multiple methods of correct recognition” is likely quantifying how perceptually unique a given phoneme string is and serves as a more flexible measure of a particular stimulus word’s intelligibility.

**Measuring perceptual confusability**

When the number of lexical entries that are allowed to compete is held constant (at all monosyllabic words), calculating similarity using HMMs accounted for only marginally significant unique variance beyond the original NAM method. Given that these measures are constructed from the same confusion matrices and are highly correlated, the only very small difference between HMM and NAM-derived measures may not be surprising. The key difference between the HMM and NAM measures is that the HMM measures produce scores that satisfy the mathematical definition of a probability function; HMMs also allow for multiple ways for the words to be confused, rather than considering only the line-up-the-vowel path to confusion. However, particularly for CVs, the line-up-the-vowel method is likely to render the largest values for most word pairs; the “cast|cat” situation is relatively rare in the lexicon. However, as words get more complex, they will provide more opportunities for complex confusions. Longer words tend to have fewer neighbors, making DAS measures generally less informative. However, Suárez et al. (2011) demonstrated effects of lexical competition from nearby words even for targets with no direct DAS neighbors (see discussion of Levenshtein distance below). Future research should evaluate whether our HMM-based quantifications of lexical competition predict recognition accuracy for longer words more accurately than Levenshtein distance metrics in the style of Suárez et al. (2011).

**Quantifying the spread of lexical activation**

The third analysis revealed that a larger subset of the lexicon is activated during spoken-word recognition than the DAS shortcut method would predict. Although the original implementation of the NAM included all monosyllabic words as potential competitors for each stimulus word, it is common in the literature to quantify lexical competition using only words within a one-phoneme radius of the stimulus word. Given that adding the influence of conf_all significantly improved the model after controlling for conf.CV, conf_mono, and conf.DAS, the results indicate that even perceptually distant words are simultaneously activated and therefore provide competition for stimulus words. The finding that adding any of the other confusability measures to conf.CV improves fit also provides further empirical support that words of differing CV structures also provide competition, as would be predicted by the NAM.

The finding that words that are several phonemes removed from the target influence recognition complements work using phonological Levenshtein distance (Suárez et al., 2011) that demonstrated that lexical
competition effects can emerge even for words with no
direct DAS neighbors. Phonological Levenshtein distance
is typically calculated as the average number phoneme
substitutions, additions, or deletions required to turn the
target word into its 20 closest neighbors in a lexicon
(Suárez et al., 2011). Measures of phonological Levenshtein
distance assign an equivalent “cost” to each phoneme
change while HMMs calculate the differences between a
word and a competitor continuously. However, the predict-
power of phonological Levenshtein distance and the
current HMM work suggest that a larger subset of the
lexicon is activated during spoken-word recognition than
the DAS shortcut method implicitly assumes. Although
measures of Levenshtein distance quantify lexical competi-
tion more sensitively than simply counting DAS neighbors,
HMMs are able to quantify the effects of even more distant
competitors, which the current results suggest may be
informative. In addition, HMMs avoid specifying an
arbitrary competitor count (20 words) that makes up the
competitor set. Given that orthographic Levenshtein dis-
tance predicts visual word recognition accuracy (Yarkoni,
Balota, & Yap, 2008), future studies should seek to evaluate
whether HMMs could also be applied to model visual word
recognition.

The finding that perceptually distant words affect
recognition is at odds with the Shortlist model (Norris,
1994), which posits that the process of lexical competition
occurs within a small (“shortlist”) of lexical items that
closely match the bottom-up input. TRACE (McClelland &
Elman, 1986) and PARSYN (Luce et al., 2000), however,
propose that any word candidates that match a portion
of the speech input are activated. Therefore, these models
could, in principle, account for our analyses’ suggestion
that words that are perceptually distant from the target
word and a competitor continuously. However, the predic-
tive power of phonological Levenshtein distance and the
current HMM work suggest that a larger subset of the
lexicon is activated during spoken-word recognition than
the DAS shortcut method implicitly assumes. Although
measures of Levenshtein distance quantify lexical competi-
tion more sensitively than simply counting DAS neighbors,
HMMs are able to quantify the effects of even more distant
competitors, which the current results suggest may be
informative. In addition, HMMs avoid specifying an
arbitrary competitor count (20 words) that makes up the
competitor set. Given that orthographic Levenshtein dis-
tance predicts visual word recognition accuracy (Yarkoni,
Balota, & Yap, 2008), future studies should seek to evaluate
whether HMMs could also be applied to model visual word
recognition.

Evaluating the source of lexical competition

The fourth analysis revealed that the source of the
lexical competition — that is, whether competition comes
primarily from one frequent, highly similar competitor or
from a greater number of moderately similar competitors
— influences spoken-word recognition. Words that have a
higher proportion of their competition coming from a
single neighbor (those with high values for NN_ratio) were
recognized more accurately than those whose total compe-
tition is more evenly distributed across multiple neigh-
bors.4 This result suggests that recognition is facilitated
when a target word may be easily discriminated from most
other words in the lexicon and is left to compete primarily
with one highly activated competitor.

In contrast, Strand (2014) found that greater dispersion
in the distribution of competitors impaired word-
recognition accuracy after controlling for the amount of
overall lexical competition. That is, words that have a
greater proportion of their competition originating from
highly similar words are recognized less accurately than
those whose competition is more uniform across competi-
tors. That finding is seemingly difficult to reconcile with
the current result, but several methodological differences
complicate direct comparisons between the two studies:
Strand (2014) used an alternate metric for calculating
NSWPs and assessed the dispersion of the distributions,
rather than the proportion of competition from the nearest
neighbor. Future studies should seek to elucidate the
apparently complex relationship between the distribution
of competitors and word-recognition accuracy.

Although the two results seem to pull in opposite direc-
tions, both Strand (2014) and the findings reported here
agree that the shape of the distribution of competitors in
lexical space, and not just the total amount of competition,
influences word-recognition accuracy.5 Critically, metrics
that combine the influence of all competitors by a simple
sum (as FWNPs do) neglect the influence of the variation
in competitor distance. The fact that the distribution of com-
petition influences recognition is difficult to explain using
existing models of word recognition. Spoken-word recogni-
tion models such as TRACE (McClelland & Elman, 1986)
and PARSYN (Luce et al., 2000) assume that the degree of
competition or inhibition is based on all the activated lexical
representations and produce patterns of inhibitory and facil-
itative effects among competitors based on how and when
they are activated (cf. Chen & Mirman, 2012, 2015). How-
ever, they do not include a mechanism that is sensitive to
the distribution of those inputs. That is, the input integration
mechanism of the TRACE, which calculates the activation of
the target based on the activation and inhibition from other
words, does not include a mechanism for incorporating the
shape of the distribution. Although it is possible that these
effects may be represented by patterns of interactive activa-
tion among competitors that are also competitors of one
another, it is an open question whether TRACE could account
for the present findings.

Potential limitations in quantifying competition

There are several limitations to the current computa-
tions that may lead to improved predictive power if they
are addressed. Some of these limitations are purely the
result of limitations of the phoneme-confusion data, while
others are more structural.

Our first, and most persuasive, data-based limitation is
that the phoneme-confusion data from Luce & Pisoni
included null-response categories for consonants (and data
for participants’ reports of consonants even when none
was presented), but not for vowels. Thus we have no esti-
mate for the probability of perceiving a vowel when none
was presented, and therefore our vowel hallucination
probabilities are all set to zero. This zero-probability
setting means that our HMM-based measures implausibly

4 Follow-up analyses confirm that this finding is not the result of
collinearity between NN_ratio and frequency or predicted intelligibility.
5 An anonymous reviewer also points out that the two small effects that
pull in opposite directions are comparatively weak, and may be explained
more parsimoniously by claiming that there, in fact, is no effect: therefore,
future studies should attempt to replicate and expand these findings to
assess whether and how the shape of the distribution influences word
recognition.
report zero probability of confusion of any competitor with strictly more vowels than the stimulus word (e.g., “polite” for “plight”). Renrunning Analysis 3 with a better estimate of conf_all to include this missing data could shed more light on the spread of activation throughout the lexicon.

A second data-based limitation is that, in the current study, phoneme-confusion rates were derived from phoneme-identification tasks consisting of identifying individual phonemes embedded in a consistent phonemic context. However, it is likely that the patterns of confusion observed would vary in a different phonemic context, due to coarticulation or variation of perceptual salience in different acoustic contexts. Thus a single phoneme-to-phoneme confusion probability may not accurately represent the confusability of phoneme strings that appear in different contexts in different real words. As an extreme example, it may be that a given pair of vowels is very confusable when they follow a stop consonant, but if they follow a fricative consonant then they are clearly distinguishable. A similar limitation occurs for hallucinations: the current HMMs model the cost of inserting a given phoneme as identical regardless of what phonemes are adjacent. An alternative phrasing is that single phoneme identifications do not allow us to assess the similarity of individual phonemes and phoneme clusters. For example, our HMMs would compute “cast” and “cats” as equally similar to “cat.” However, it is certainly possible that “st” and “t” are more confusable than “ts” and “s.” It is an advantage over the NAM’s original formulation that HMMs can very naturally incorporate these distinctions — the emission probabilities of any particular state in the HMM could simply be defined based on that particular phoneme’s context — but the current matrix of phoneme confusions does not include them. Thus, the current phoneme confusion data for calculating competition does not enable us to test these hypotheses.

The HMMs presented here also do not capture the fact that words unfold over time. This feature of word recognition is incorporated into TRACE (McClelland & Elman, 1986), Shortlist (Norris, 1994) and Cohort (Marslen-Wilson, 1987) models, such that words that differ at onset provide less competition than those that differ at offset, a prediction that has been empirically supported (Allopenna, Magnuson, & Tanenhaus, 1998). Adapting HMMs to include this feature would require more substantial modification to the model’s infrastructure; it is not clear how to add these features to HMMs in a mathematically coherent way. Another feature of these models that cannot be readily incorporated into HMMs is phonemic variability from coarticulation with upcoming sounds; the natural way to include context is conditioned on the previous phoneme(s) and not on upcoming phoneme(s). Future studies should explore methods to include these features in the framework of the NAM.

### Clinical applications

Because the only input necessary to calculate lexical distances is phoneme-confusion matrices, it is possible to map the topology of lexical space for population groups with specific perceptual characteristics, such as cochlear implant users or people with impaired hearing. Population-specific measures of lexical distances may also inform research on word recognition in older adults. Evidence exists that older adults may be especially impaired at recognizing words from regions of the lexicon that are perceptually dense [as quantified by the DAS shortcut method (Sommers, 1996)]. However, it is possible that older adults are less able to make distinctions between phonemic contrasts than are younger adults, due to either sensory deficits (i.e., presbycusis) or age-related cognitive changes (Pichora-Fuller, 2003). Therefore, for any given stimulus word, there may be more words that serve as potential competitors for older adults than for younger adults, assuming that older adults will show impaired phoneme discrimination compared with younger adults. It may be that the observed interaction between age and lexical difficulty is due to age-related changes in lexical competition instead of (or, more likely, in addition to) age-related cognitive changes (e.g., inhibitory deficits) (see Ramscar, Hendrix, Shaoul, Milin, & Baayen, 2014 for a discussion of how the mental lexicon changes with age).

### Conclusions

The concepts of activation and competition are well supported in research on spoken-word recognition, and the NAM has proven itself to be an influential method of formalizing these concepts. The current study suggests that more flexible approaches for quantifying how activation and competition unfold may help shed light on the mechanisms underlying spoken-word recognition.

### Acknowledgments

We thank Emily Massell and Violet Brown for assistance with data collection, and Anna Rafferty for helpful discussions. This work was supported by Carleton College. Portions of this work were performed while the second author was visiting Churchill College and the Computer Laboratory at the University of Cambridge; thanks to both for their support and hospitality.

### References


