LDL-AURIS: Error-driven Learning in Modeling Spoken Word Recognition

Elnaz Shafaei-Bajestan, Masoumeh Moradipour-Tari, and R. Harald Baayen

Quantitative Linguistics, Eberhard Karls University of Tübingen, Tübingen, Germany

ARTICLE HISTORY
Compiled June 26, 2020

ABSTRACT
A computational model for auditory word recognition is presented that enhances the model of Arnold et al. (2017). Real-valued features extracted from the speech signal instead of discrete features. One-hot encoding for words' meanings is replaced by real-valued semantic vectors, adding a small amount of noise to safeguard discriminability. Instead of learning with Rescorla-Wagner updating, we use multivariate multiple regression, which captures discrimination learning in the limit of experience. These new design features substantially improve prediction accuracy for words extracted from spontaneous conversations. They also provide enhanced temporal granularity, enabling the modeling of cohort-like effects. Clustering with t-SNE shows that the acoustic form space captures phone-like similarities and differences. Thus, wide learning with high-dimensional vectors and no hidden layers, and no abstract mediating phone-like representations, is not only possible but also achieves excellent performance that approximates the lower bound of human accuracy on the challenging task of isolated word recognition.

KEYWORDS
Spoken word recognition; error-driven learning; Rescorla-Wagner learning rule; Widrow-Hoff learning rule; Naive Discriminative Learning; Linear Discriminative Learning; multivariate multiple regression

1. Introduction

We present a computational model for auditory word recognition that enhances the model of Arnold et al. (2017). Several aspects of their and our approach to auditory word recognition are of special interest from a cognitive perspective. First, the acoustic features of our model are inspired by the signal pre-processing that takes place in the cochlea. Second, the model does not need the huge and cognitively implausible volumes of data for training required by current deep learning networks. Third, the model can handle the huge variation in the way in which words are pronounced, and does so with a simple network that has no hidden layers. The endstate of incremental discriminative learning with this network is formally equivalent to that obtained with multivariate multiple regression applied to all data points at once. We show that our revised model affords substantially improved prediction accuracy for words extracted from spontaneous conversational speech. The revised model also provides enhanced temporal granularity, enabling the modeling of cohort-like effects. Clustering with t-

CONTACT Elnaz Shafaei-Bajestan. Email: elnaz.shafaei-bajestan@uni-tuebingen.de

SNE shows that the acoustic form space defined by the revised acoustic features captures phone-like similarities and differences. Thus, wide learning with high-dimensional vectors and no hidden layers, and no abstract mediating phone-like representations, is not only possible but also achieves excellent performance that approximates the lower bound of human accuracy on the challenging task of isolated word recognition.

In what follows, we first provide an overview of previous models of auditory word recognition. We then present central computational details of our model, and discuss its performance evaluated on 131,000 auditory tokens of English words extracted from the UCLA Library Broadcast NewsScape corpus, a massive library of audiovisual TV news recordings. In the general discussion, we return to the implications of our approach for linguistic and cognitive theories of the lexicon and lexical processing.

2. Architectures for spoken word recognition

In linguistics, the hypothesis of the duality of patterning of language has attained axiomatic status. Language is considered to be a symbolic system with a two-level structure. One level concerns how meaningless sounds pattern together to form meaningful units, the words and morphemes of a language. The other level is concerned with the calculus of rules that govern how words and morphemes can be assembled combinatorially into larger ensembles (Chomsky and Halle, 1968; Hockett and Hockett, 1960; Licklider, 1952; Martinet, 1967).

Accordingly, most cognitive models of spoken word recognition (henceforth SWR) such as the TRACE (McClelland and Elman, 1986), the COHORT (Marslen-Wilson, 1987), the SHORTLIST (Norris, 1994a), the Neighborhood Activation Model (Luce et al., 2000), the SHORTLIST-B (Norris and McQueen, 2008), and the FINE-TRACKER (Scharenborg, 2008) all posit two levels of representation and processing, a lexical and a prelexical level. The prelexical level is put forward to enable the system to convert the continuous varying input audio signal into discrete non-varying abstract units, sequences of which form the lexical units functioning in the higher-level combinatorics of morphology and syntax. The main motivation for having an intermediate phone-based representation is that phones are judged to be crucial for dealing with the huge variability present in the speech signal. Thus, at the prelexical level, the speech signal is tamed into phones, and it is these phones that can then be used for lexical access (Diehl et al., 2004; McQueen, 2005; Norris and McQueen, 2008; Phillips, 2001).

Although traditional SWR models posit a prelexical level with a finite number of abstract phone units, the psychological reality of an intermediate segmental level of representation has been long debated (see Pisoni and Luce (1987) for a review, and Port and Leary (2005) for linguistic evidence). Furthermore, the exact nature of these phone units is admittedly underspecified (McQueen, 2005). Unsurprisingly, SWR models define their prelexical representation in very different ways. SHORTLIST and SHORTLIST-B work with phones and phone probabilities, TRACE posits multi-dimensional feature detectors that activate phones, and FINE-TRACKER implements articulatory-acoustic features. Unfortunately, most models remain agnostic on how their prelexical representations and phone units can actually be derived from the speech signal. As observed by Scharenborg and Boves (2010),

> "the lack of a (cognitively plausible) process that can convert speech into prelexical units not only raises questions about the validity of the theory, but also complicates attempts to compare different versions of the theory by means of computational modelling experiments."

Nevertheless, many modelers assume that some intermediate phone level is essential. Dahan and Magnuson (2006), for instance, motivates the acceptance of a prelexical level by the theoretical assumption that separation of tasks in a staged double-layered system engenders cognitive *efficiency* because of the restrictions imposed on the amount of information available for smaller mappings at each stage. The only model that argues against a mediating role of phones is the Distributed Cohort Model (Gaskell and Marslen-Wilson, 1997, 1999), which is motivated in part by the experimental research of Warren (1970, 1971, 2000), which provides evidence that the awareness of phonemes is a post-access reconstruction process.

In many SWR models, word meanings, the ultimate goal of lexical access (Harley, 2013), are represented at a dedicated lexical layer. A review of the different ways in which meanings have been represented in the literature is given by Magnuson (2017), here, we focus on two dominant approaches.

Firstly, in localist approaches, as implemented by the logogen model (Morton, 1969), TRACE, SHORTLIST, FINE-TRACKER, Neighborhood Activation Model, PARSYN (Luce et al., 2000), and DIANA (ten Bosch et al., 2015), the mental lexicon provides a list of lexical units that are either symbolic units or unit-like entries labelled with specifications of the sequence of phones against which the acoustic signal has to be matched. Once a lexical unit has been selected, it then provides access to its corresponding meaning.

Secondly, in distributed approaches, adopted by models such as Distributed Cohort Model and EARSHOT (Magnuson et al., 2018), a word's meaning is represented by a numeric vector specifying the coordinates of that word in a high-dimensional semantic space. The status of phone units within these approaches is under debate. The Distributed Cohort Model argues that distributed recurrent networks obviate the need for intermediate phone representations, and hence this model does not make any attempt to link patterns of activation on the hidden recurrent layer of the model to abstract phones. By contrast, the deep learning model of Magnuson et al. (2018) explicitly interprets the units on its hidden layer as the fuzzy equivalents in the brain of the discrete phones of traditional linguistics.

These models of SWR provide theories of the mental lexicon that have several problematic aspects. First, the input to most models of auditory word recognition is typically a symbolic approximation of real conversational speech. The only models that work with real speech are FINE-TRACKER (Scharenborg, 2008, 2009), DIANA, and EARSHOT (Magnuson et al., 2018). Of these models, FINE-TRACKER and DIANA are given clean laboratory speech as input, whereas EARSHOT limits its input to a list of 1000 words generated by a text-to-speech system. However, normal daily conversational speech is characterized by enormous variability, and the way in which words are produced often diverges substantially from their canonical pronunciation. For instance, a survey of the Buckeye corpus (Pitt et al., 2005) of spontaneous conversations recorded at Columbus, Ohio (Johnson, 2004) revealed that around 5% of the words are spoken with one syllable missing, and that a little over 20% of words have at least one phone missing (see also Ernestus, 2000; Keune et al., 2005). It is noteworthy that adding entries for reduced forms to the lexicon has been shown not to afford better overal recognition (Cucchiarini and Strik, 2003). Importantly, canonical forms do not do justice to how speakers modulate fine phonetic detail to fine-tune what they want to convey (Hawkins, 2003). Plag et al. (2017) have documented that the acoustic duration of word final [s] in English varies significantly in the mean depending on its inflectional function (see also Tomaschek et al., 2019). Thus, if the speech signal were to be reduced to just a sequence of canonical phones, then large amounts of information present in

the speech signal would be lost completely. As a consequence, models of SWR have to take on the challenge of taking real spontaneous speech as input.

Second, models of SWR typically do not consider how their parameter settings (including parameters such as the number of hidden layers) are learned. Models such as TRACE and SHORTLIST make use of connection weights that are fixed and set by hand. The Bayesian version of SHORTLIST estimates probabilities on the basis of a fixed corpus. The parameters of the Hidden Markov Model underlying DIANA are likewise tuned by hand and then frozen. Connectionist models such as Distributed Cohort Model and EARSHOT are trained incrementally, and hence can be considered learning models. In practise, these models are trained until their performance is deemed satisfactory, after which the model is taken to characterize an adult word recognition system. However, vocabulary size is known to increase over the lifetime (Keuleers et al., 2015; Ramscar et al., 2014) and ideally the dynamics of life-long learning should be part of a learning model of lexical processing. By contrast, current deep learning models typically require many passes through the training data, and training is typically terminated when a sweet spot has been found where prediction accuracy under cross-validation has reached its maximum. Since further training would lead to a reduction in accuracy, training is terminated and no further learning can take place. Importantly, when small and/or simplified datasets are used for training, they can easily overfit the data and may not generalize well.

Third, all the above models work with a fixed lexicon. When morphologically complex words are included, as for instance in SHORTLIST-B, no mechanisms are implemented that would allow the model to recognize out-of-vocabulary inflected or derived words that have in-vocabulary words as their base words. In other words, these models are all full-listing models (Butterworth, 1983).

In what follows, we present a cognitively motivated, mathematically well-understood, computationally implemented model for SWR that avoids the above three theoretical problems and that, trained on real conversational speech, shows excellent performance. As this model builds on previous modeling work using Naive Discriminative Learning (NDL; Baayen et al., 2011) and Linear Discriminative Learning (LDL; Baayen et al., 2019), the next section provides an introduction to these modeling approaches. The subsequent section then describes the changes we implemented in order to improve both model performance for SWR and to make the model cognitively more plausible.

3. Previous modeling of SWR with NDL and LDL

3.1. Informal characterization of NDL and LDL

The NDL and LDL models are grounded in error driven learning as formalized in the learning rules of Rescorla and Wagner (1972) and Widrow and Hoff (1960). These two learning rules are closely related, and are actually identical under specific parameter settings. As we shall see below, both implement a form of incremental multiple linear regression, and both rules can be also seen as simple artificial neural networks with an input layer with cues, an output layer with outcomes, and no hidden layers (the terminology of cues and outcomes is borrowed from Danks (2003)). Cues are sublexical form features, and outcomes are values on the axes of a high-dimensional semantic space. Error-driving learning as formalized by Rescorla and Wagner (1972) has proven to be fruitful for understanding both animal learning (Bitterman, 2000; Gluck and

Myers, 2001; Rescorla, 1988) and human learning (Ellis, 2006; Nixon, 2020; Olejarczuk et al., 2018; Ramscar et al., 2014; Ramscar and Yarlett, 2007; Ramscar et al., 2010; Siegel and Allan, 1996).

Statistically, a model trained with the Rescorla-Wagner learning rule is a classifier that is trained to predict whether or not a specific outcome is present. Naive discriminative learning extends this single-label classifier to a multiple-label classifier by having the model learn to predict multiple outcomes in parallel. For instance, Baayen et al. (2011) built a model for Serbian case-inflected nouns, and for the noun *ženama* taught the model to predict three labels (classes): WOMAN, PLURAL, and DATIVE (see Sering et al., 2018, for mathematical details). Naive discriminative learning has been used successfully for modeling, e.g., unprimed visual lexical decision latencies for both simple and morphologically complex words (Baayen et al., 2011, 2016a), masked priming (Milin et al., 2017), non-masked morphological priming (Baayen and Smolka, 2020), and the acoustic duration of English syllable-final *s* (Tomaschek et al., 2019).[1] Arnold et al. (2017) and Shafaei-Bajestan and Baayen (2018) used naive discriminative learning to train multiple-label classifiers for SWR for German and English respectively. Both models made use of cues that were extracted from the audio signal. Below, we discuss how this was done in further detail, and in this study we will show how their method of signal preprocessing can be enhanced. Importantly, both studies took as input the audio files of words extracted from spontaneous conversational speech.

Linear Discriminative Learning (Baayen et al., 2019) relaxes the assumption made by Naive Discriminative Learning that outcomes are coded as present (1) or absent (0). By allowing outcomes to be real numbers, words' meanings can now be represented using vector representations from distributional semantics (Landauer and Dumais, 1997; Mikolov et al., 2013). Mathematically, LDL models are equivalent to multivariate multiple regression models. Baayen et al. (2019) tested their model on 130,000 words extracted from 20 hours of speech sampled from the UCLA Library Broadcast NewsScape data. Chuang et al. (2020) trained an LDL model on the audio files of the MALD database (Tucker et al., 2017), and used this model to predict the acoustic durations and auditory lexical decision latencies to the auditory nonwords in this database.

The Rescorla-Wagner and Widrow-Hoff learning rules implement incremental error-driven learning that uses gradient descent (for mathematical details, see the next section). Alternatively, one can estimate the 'endstate' or 'equilibrium state' of learning. This endstate provides the connection strengths between cues and outcomes for an infinite number of tokens sampled from the training data. Danks (2003) provides equilibrium equations for the endstate of learning with the Rescorla-Wagner learning rule. As shown by Baayen et al. (2019), LDL is mathematically equivalent to multivariate multiple regression. Table 1 provides an overview of how NDL and LDL set up representations and error-driven learning. In the next section, we provide details on the mathematics underlying NDL and LDL. Readers who are not interested in the technical details, can proceed to section 3.3.

3.2. Formal model definitions

The error-driven learning algorithms of Rescorla-Wagner, Widrow-Hoff, and LDL regression are supervised learning algorithms that learn the weights on the connections between cue (input) and outcome (output) values in single-layer artificial neural net-

[1]For details on how NDL and LDL model the processing of morphologically complex words, see Baayen et al. (2018), Chuang et al. (2019), and Baayen and Smolka (2020).

Table 1. Overview of NDL and LDL

	NDL	LDL
cues	discrete (1/0)	discrete (1/0)
outcomes	discrete (1/0)	real-valued
incremental learning	Rescorla-Wagner	Widrow-Hoff
endstate of learning	Danks equilibrium equations	multivariate multiple regression

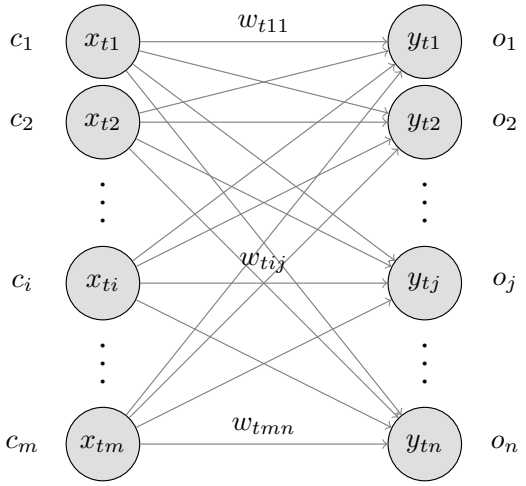

Figure 1. A one-layer fully-connected feed-forward neural network during learning at trial t.

works with as objective to minimize the discrepancy between the desired outcome and the system's predicted outcome. The first two models achieve this mapping by updating the weights step by step as learning events are presented to the model. The third algorithm calculates the final state of learning using the matrix algebra of multivariate multiple regression. We begin with formally defining the task of iterative learning from a training set.

Definition 3.1. (learning).
Given

- scalars m, n, and p,
- a set $C = \{c_i\}$ for $i \in [\,1 \cdots m\,]$, where c_i is a cue,
- a set $O = \{o_j\}$ for $j \in [\,1 \cdots n\,]$, where o_j is an outcome,
- a set $X = \{\boldsymbol{x}_t\}$ for $t \in [\,1 \cdots p\,]$, where $\boldsymbol{x}_t$ is a row vector over C,
- a set $Y = \{\boldsymbol{y}_t\}$ for $t \in [\,1 \cdots p\,]$, where $\boldsymbol{y}_t$ is a row vector over O,
- a labeled training sequence of learning events $T = (e_t)$ for $t \in [\,1 \cdots p\,]$, where $e_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$ is a learning event,

compute a mapping $P : X \to Y$ such that $P(\boldsymbol{x}_t) \approx \boldsymbol{y}_t$.

We first consider incremental learning. Here, we use a single-layer fully-connected feed-forward network architecture for learning the mapping P from the training sequence T (Figure 1). This network has m neurons in the input layer, n neurons in the output layer with activation function f, and $m \times n$ connections from the input layer to the output layer. Input vector $\boldsymbol{x}_t = [x_{ti}]_{1\times m}$ stores x_{ti}, the value that input neuron c_i assumes at trial t, and output vector $\boldsymbol{y}_t = [y_{tj}]_{1\times n}$ stores y_{tj}, the value that output neuron o_j assumes at trial t. The weight on the connection from c_i to o_j at trial t is denoted as w_{tij}.

At trial t, an output neuron o_j receives m input values x_{ti} on afferent connections with associated weights w_{tij}, and combines the input values into the net input activation a_{tj}

$$a_{tj} = \sum_{i=1}^{m} x_{ti} w_{tij}.$$

In neural networks, a variety of activation functions f are available for further transforming this net input activation. In our model, f is always the identity function, but f can be chosen to be any Riemann integrable function. Thanks to using the identity function, in our model, the neuron's predicted output $\hat{y}_{tj}$ is simply

$$\hat{y}_{tj} = f(a_{tj}) = a_{tj}.$$

The error for a neuron is defined as the difference between the desired target output and the output produced by the neuron:

$$E_{tj} = y_{tj} - \hat{y}_{tj}.$$

The error for the whole network is defined as sum of squared residuals divided by two:

$$E_t = \sum_{j=1}^{n} \frac{1}{2}(y_{tj} - \hat{y}_{tj})^2. \tag{1}$$

Both the Rescorla-Wagner and the Widrow-Hoff learning rules try to find the minimum of the function E_t using gradient descent (Hadamard, 1908), an iterative optimization algorithm for finding a local minimum of a function. The algorithm, at each step, moves in the direction of the steepest descent at that step. The steepest descent is defined by the negative of the gradient. Therefore, at each trial, it changes the weight on the connection from c_i to o_j,

$$w_{tij} = w_{(t-1)ij} + \Delta w_{tij},$$

proportional to the negative of the gradient of the function E_t:

$$\Delta w_{tij} \propto -\frac{\partial E_t}{\partial w_{tij}}.$$

Thus, assuming a constant scalar η, often referred to as the learning rate, the changes in weights at time step t are defined as $\Delta w_{tij} = -\eta \frac{\partial E_t}{\partial w_{tij}}$ or,

$$\Delta w_{tij} = \eta(y_{tj} - \hat{y_{tj}}) f'(a_{tj}) x_{ti} \tag{2}$$

(see appendix A.1 for a proof of (2), which is known as the Delta rule and as the Least Mean Square rule). After visiting all learning events in the training sequence, the state of the network is given by a weight matrix $\boldsymbol{W} = [w_{(t=p)ij}]_{m \times n} = [w_{ij}]_{m \times n}$. The requested mapping P in definition 3.1 is given by element-wise application of the activation function f to the net input activations. Since in our model, f is the identity

function, we have that

$$P(\boldsymbol{x}_t) = f \circ (\boldsymbol{x}_t \boldsymbol{W}) = \boldsymbol{x}_t \boldsymbol{W} = \hat{\boldsymbol{y}}_t.$$

Widrow-Hoff learning assumes that $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ are real-valued vectors in $\mathbb{R}^m$ and $\mathbb{R}^n$ respectively. Rescorla-Wagner learning is a specific case of the general definition of 3.1 in which the cues and outcomes of the model can only take binary values, representing the presence or absence of discrete features in a given learning event. Rescorla-Wagner also restricts the activation function f to be the identity function (see appendix A.2 for further details).

Instead of building up the network incrementally and updating the weights for each successive learning event, we can also estimate the network, or its defining matrix $\boldsymbol{W}$ in one single step, taking all training data into account simultaneously. To do so, we take all learning trials together by stacking the input vectors $\boldsymbol{x}_t$ in matrix $\boldsymbol{X} = [x_{ti}]_{p \times m}$ and stacking the output vectors $\boldsymbol{y}_t$ in matrix $\boldsymbol{Y} = [y_{tj}]_{p \times n}$, for all i, j, t. We are interested in finding a mapping that transforms the row vectors of $\boldsymbol{X}$ into the row vectors of $\boldsymbol{Y}$ as accurately as possible. Here, we can fall back on regression modeling. Analogous to the standard multiple regression model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

we can define a multivariate multiple regression model

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} + \boldsymbol{E} \tag{3}$$

with errors $\boldsymbol{\varepsilon}$ and $\boldsymbol{E}$ being i.i.d. and following a Gaussian distribution. The multivariate regression model takes a multivariate predictor vector $\boldsymbol{X}_{t.}$, weights each predictor value by the corresponding weight in $\boldsymbol{B}_{.t}$, resulting in a vector of predicted values $\hat{\boldsymbol{Y}}_{t.}$. Assume that $\boldsymbol{X}$ is an m-by-m square matrix with determinant $\det(\boldsymbol{X}) \neq 0$. Then there exists a matrix $\boldsymbol{X}^{-1}$, the inverse $\boldsymbol{X}$, such that

$$\boldsymbol{X}\boldsymbol{X}^{-1} = \boldsymbol{X}^{-1}\boldsymbol{X} = \boldsymbol{I}_m,$$

where $\boldsymbol{I}_m$ is the identity matrix of size m. Then, the matrix of coefficients $\boldsymbol{B}$ is given by

$$\boldsymbol{B} = \boldsymbol{X}^{-1}\boldsymbol{Y}$$

(see appendix A.3 for illustration). In practice, $\boldsymbol{X}$ is singular, i.e., its determinant is 0, and the inverse does not exist. In this case, the Moore-Penrose (Penrose, 1955) generalized matrix inverse X^+ can be used

$$\boldsymbol{B} = \boldsymbol{X}^{+}\boldsymbol{Y}.$$

Calculating the Moore-Penrose pseudoinverse is computationally expensive, and to optimize calculations the system of equations $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B}$ can be recast as

$$\begin{aligned} \boldsymbol{Y} &= \boldsymbol{X}\boldsymbol{B} \\ (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{B} \\ (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} &= \boldsymbol{B}. \end{aligned} \tag{4}$$

The inverse is now required for the smaller matrix $\boldsymbol{X}^T\boldsymbol{X}$. In this study, we estimate $\boldsymbol{B}$ using the Moore-Penrose pseudoinverse. (Another method that we are currently implementing for solving $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B}$ is Choleski decomposition.) Returning to our model for SWR, we replace the multivariate multiple regression equation (3) with

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{W}, \tag{5}$$

where $\boldsymbol{W}$, the matrix defining the connection weights in the network, replaces the matrix of coefficients $\boldsymbol{B}$. We will show below that $\boldsymbol{W}$ provides us with a network that has reached the endstate of learning, where its performance accuracy is maximal.

The twin models of NDL and LDL can now be characterized mathematically as follows. NDL's incremental engine uses Rescorla-Wagner,LDL's incremental engine is Widrow-Hoff.[2] For the endstate of learning, NDL uses the equilibrium equations of Danks (2003), which yield exactly the same weight matrix as the one obtained by solving (5). LDL's endstate model uses the multivariate multiple regression model using (5).

Given a trained model with weight matrix $\boldsymbol{W}$, the question arises of how to evaluate the model's predictions. For a learning event e_t, NDL returns the outcome o_j with the highest value in the predicted outcome vector:

$$\underset{o_j}{\operatorname{argmax}}\, \boldsymbol{x}_t\boldsymbol{W}.$$

LDL calculates the Pearson correlation coefficients of the predicted outcome vector $\hat{\boldsymbol{y}}_t$ and all gold standard outcome vectors $\boldsymbol{Y}_t$, resulting in a vector of correlations $\boldsymbol{r}_t = [\, r(\hat{\boldsymbol{y}}_t, \boldsymbol{Y}_t)\,]_{1\times p}$, and returns the word type for the token with the highest correlation value

$$\underset{\boldsymbol{y}_t}{\operatorname{argmax}}\, \boldsymbol{r}_t.$$

3.3. Challenges for Spoken Word Recognition with LDL

The two studies using LDL for auditory word recognition, Baayen et al. (2019) and Shafaei-Bajestan and Baayen (2018), report good accuracy on the training data, but the latter study documents that accuracy is halved under cross-validation (but still superior to that of Mozilla Deep Speech [3]). It is therefore possible that LDL is substantially overfitting the data and that its cross-validation accuracy is by far not as good as its accuracy on the training data. To place this question in perspective, we first note that models for visual word recognition as well as models such as TRACE and SHORTLIST, have worked with invariable symbolic input representations for words' forms. However, in normal conversational speech, the wave forms of different tokens of the same linguistic word type are never identical, and often vary substantially. Thus, whereas models working with symbolic representations can dispense with cross-validation, models that take real speech as input cannot be evaluated properly without cross-validation on unseen acoustic tokens of known, previously encountered, linguistic word types.

[2] For further details and optimized code for incremental learning, including also the Kalman filter, see Milin et al. (2020).

[3] See `https://github.com/mozilla/DeepSpeech` (last accessed June 26, 2020).

A related issue is whether LDL, precisely because it works with linear mappings, may be too restricted to offer the desired accuracy under cross-validation. Thus, it is an empirical question whether the hidden layers of deep learning can be truly dispensed with. If hidden layers are indeed required, then this would provide further support for the position argued for by (Magnuson, 2017) that phonemes are essential to SWR and that they emerge naturally in a deep learning network's hidden layers (but see Gaskell and Marslen-Wilson (1997, 1999) for counterevidence). Furthermore, even if LDL were to achieve good performance under cross-validation, using a linear mapping from acoustic features to semantic vectors, then how would the model account for the evidence for phone-like topological maps in the cortex (see, e.g. Cibelli et al., 2015)?

There is one other aspect of the question concerning potential overfitting that requires further investigation. It is well known that also deep learning networks run the risk of overfitting. Often there is a sweet spot as the model is taken through the dataset repeatedly to optimize its weights, at which accuracy under cross-validation reaches its maximum. With further training, accuracy on the training set then increases, whereas accuracy under cross-validation decreases — the hallmark of overfitting. Weitz (2019) observed that the loss function for an LSTM network classifying between 100 word types of our data set repeatedly jolts sharply out of local minimum beyond a threshold for training. This raises the question of whether the endstate of learning, as used by LDL, is actually optimal when accuracy is evaluated under cross-validation. If it is suboptimal, then incremental learning in combination with cross-validation is preferable under the assumption that there is a sweet spot where accuracy on the training data and on unseen data are properly balanced.

A very different challenge to NDL and LDL comes from classical cognitive models of SWR that provide predictions over time for the support that a target word and its closest competitors receive from the incoming auditory stimulus (Marslen-Wilson, 1984), irrespective of whether words are considered in isolation as in TRACE or in sentence context (SHORTLIST). The temporal granularity of the previous NDL and LDL models (Arnold et al., 2017; Baayen et al., 2019; Shafaei-Bajestan and Baayen, 2018), however, is too coarse to be able to provide detailed predictions for cohort-like effects. An important goal of the present study is to develop enhanced acoustic features that enable the model to predict the time-course of lexical processing with greater precision.

A final challenge that we address in this study is whether further optimization of model performance is possible by enhancing the representation of words' meanings. Whereas models such as Distributed Cohort Model and EARSHOT assign randomly-generated semantic representations to words, and DIANA uses localist representations for word meanings, Baayen et al. (2019) and Chuang et al. (2020) made use of semantic vectors (aka word embeddings, see Gaskell and Marslen-Wilson, 1999, for a similar approach) derived from the TASA corpus (Ivens and Koslin, 1991; Landauer et al., 1998) using the algorithm described in Baayen et al. (2019, 2016a).[4] The TASA corpus, which with 10 million words is very small compared to the volumes of texts that standard methods from machine learning such as word2vec (Mikolov et al., 2013) are trained on (typically billions of words). Although our TASA-based semantic vectors perform well (see Long, 2018, for an explicit comparison with word2vec), they may not be as discriminable as desired, thereby reducing model performance. We therefore investigated several ways in which the choice of semantic vectors affects model performance.

[4]For the importance of working with empirical semantic vectors in computational modeling studies, see Heitmeier and Baayen (2020).

In what follows, we first address the issues surrounding potential overfitting (section 5). We then introduce enhanced acoustic features that afford greater temporal granularity (section 6). The question of what semantic representations are optimal is investigated in section 7. Section 8 brings the results from the preceding sections together and defines and tests our enhanced model for SWR, to which we will refer as LDL for AUditory word Recognition from Incoming Spontaneous speech (LDL-AURIS).

4. Data

The data used in the current study is a subset of the *UCLA Library Broadcast NewsScape* corpus[5], a massive library of audiovisual TV news recordings along with the corresponding closed captions. The subset is from 2016, consists mainly of US-American TV news and talk shows, and includes 500 audio files that are successfully aligned with their closed captions for greater than or equal to 97% of their audio words tokens using the Gentle[6] forced aligner. The aligner provides alignment at word and phone level. Subsequently, we automatically extracted the relatively clean 30-second long audio stretches where there is speech with little background noise or music, following Shafaei-Bajestan and Baayen (2018). 2287 of such segments were randomly sampled to comprise the final data set with 20 h of audio.

This data set contains 131,372 uninflected non-compound word tokens of 4741 word types. All words are lower-cased and stop words are retained. The left panel of figure 2 shows that words in this data set follow Zipf's law. Duration of audio word tokens add up to a total of 9.3 h with an average word duration of 254 ms (SD = 154, range: $10 - 1480$). One-third of the tokens are approximately between 100 to 200 ms long. The shortest audio tokens are 10 ms long and belong to the occurrences of words 'are', 'a', 'i', 'or', 'oh', 'eye', 'e', 'owe', 'o'. The longest audio token belongs to an occurrence of the word 'spectacular'. The right panel of figure 2 shows that word duration has an approximately lognormal distribution, an observation in accordance with previous findings for the distribution of American-English spoken word lengths (French et al., 1930; Herdan, 1960). This data set is employed in all simulations presented throughout the present study. All models are trained and tested on single word tokens as given by the word boundaries provided by the aligner.

The choice to model isolated word recognition is motivated primarily by the practical consideration that modeling word recognition in continuous speech is a hard task, and that a focus on isolated word recognition makes the task more manageable. This is not to say that isolated word recognition is a simple task. Pickett and Pollack (1963); Pollack and Pickett (1963) demonstrated long ago that spoken words isolated from conversational speech are difficult to recognize for human listeners: American English speech segments comprising one word with mean duration of approximately 20 cs are, on average, correctly identified between 20% to 50% of the times by native speakers, depending on speaking rate. Arnold et al. (2017) reported similar recognition accuracy percentages from 20% to 44% for German stimuli with average duration of 23 cs. Interestingly, deep learning networks are severely challenged by the task of isolated word recognition. Arnold et al. (2017) reported that the Google Cloud Speech API correctly identified only 5.4% of his stimuli. Likewise, Shafaei-Bajestan and Baayen (2018) found that that Mozilla Deep Speech, an open source implementation of a state-of-the-art

[5]See `http://newsscape.library.ucla.edu/` and `http://tvnews.library.ucla.edu/` (last accessed June 26, 2020).

[6]See `http://lowerquality.com/gentle` (last accessed June 26, 2020).

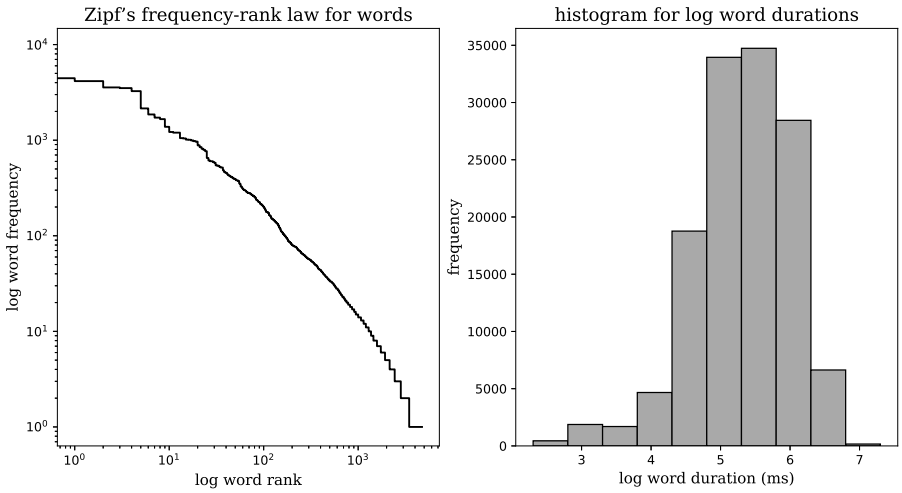

Figure 2. The data set is a good representative of the English language. The left panel shows that word frequency decreases linearly with Zipf word rank in a double logarithmic plane, a necessary condition for a power law relation. The right panel shows that word duration follows a lognormal distribution.

speech recognition system, performed with an accuracy around 6%, lagging behind the accuracy of her NDL model with around 6–9%.

A further reason for focusing at this stage of model development on isolated words is that, with the exception of the SHORTLIST models (Norris and McQueen, 2008; Norris, 1994b), computational models have also addressed single word recognition. It has been argued that recognizing individual words in utterances is a precondition for understanding (Weber and Scharenborg, 2012) (but see Baayen et al., 2016b, for a different approach).

By focusing on isolated word recognition, we are also setting ourselves the task to clarify how much information can be extracted from words' audio signals. Deep learning models for speech recognition depend heavily on language models, and current deep learning implementations may, given the abovementioned results, underestimate the mileage that can be made by careful consideration of the rich information in the acoustic signal. It is noteworthy that it has been argued that in human SWR the acoustic input has overwhelming priority (Gaskell and Marslen-Wilson, 2001; Magnuson, 2017).

5. Learning with Rescorla-Wagner, Widrow-Hoff, and Multivariate linear regression

The aim of this section is to clarify how incremental learning and the endstate of learning compare. Of specific interest is whether the endstate of learning is suboptimal compared to some intermediate stage reached through incremental learning. We also consider how working with discrete semantic vectors (with sparse binary coding of the presence of lexemes) as opposed to real-valued semantic vectors (word embeddings) affects results.

5.1. Method

For incremental learning with gradient descent training, we need to specify a learning rate (set to $\eta = 0.001$ in our simulations) and the number of iterations n through the data (set to $n = 1$ in previous studies using NDL, but varied in the present simulations). There is no need for choosing η and n when using the matrix inversion technique for estimating the endstate of learning. Inversion of large matrices can become prohibitively slow for very large datasets. Fortunately, there have been major developments in optimizing the algorithms for computations of the pseudo-inverse in computer science (see Horata et al., 2011; Lu et al., 2015, for example), and for the present data, all inverse matrices are straightforward to calculate.

Table 2 summarizes the four set-ups that we considered by crossing training method (incremental vs. endstate of learning) with the method for representing word meanings (NDL vs. LDL). For all simulations, the acoustic input is represented by the FBSF developed by Arnold et al. (2017). For incremental learning with gradient descent for Rescorla-Wagner and Widrow-Hoff we made use of the Python library `pyndl`.

The input to `pyndl` is a sequence of learning events consisting of a set of cues and a set of outcomes. For NDL, the set of outcomes provides identifiers for the lexomes realized in the speech signal. Lexomes are defined as identifiers of, or pointers to, distributional vectors for both content words and grammatical functions such as PLURAL and PAST (Baayen et al., 2016a; Milin et al., 2017). Mathematically, the set of outcome lexomes is represented by means of a binary vector with bits set to 1 for those lexomes that are present in the word, and set to 0 for all other words (see Baayen and Smolka, 2020, for further details). Since in the present study we only consider uninflected monomorphemic words and uninflected derived words with a monomorphemic base word, and no compounds words (see Baayen et al., 2019, for further details), the set of lexomic outcomes reduces to an identifier for a word's content lexome, and the corresponding semantic vector reduces to a vector with only 1 bit on (one-hot encoding). For the present data set, the lexomic vectors have a dimensionality of 4741.

For simulations using Widrow-Hoff or LDL, one-hot encoding is replaced by real-valued semantic vectors (word embeddings). The word embeddings are supplied to `pyndl` in the form of a matrix. The word identifier for the lexome in the standard input for `pyndl` is used to extract the appropriate semantic vector from this matrix. The semantic vectors that we consider here are obtained from a subset of semantic vectors derived from the TASA corpus as described in Baayen et al. (2019), comprising 12,571 word types, each of which is associated with a real-valued vector of length 4609. Henceforth, we will refer to this semantic space as the TASA1 space. TASA1 space contains vectors for all of the word types in our data set.

For both NDL and LDL, we need a matrix specifying the acoustic features for each of the word tokens in our dataset. From the features extracted for a word from the audio signal, following (Arnold et al., 2017), an input form vector is constructed with 1s for those acoustic features that are present in the word and 0s for those features that are not realized in the word. The form vectors for the data set (N =131,372) have a dimensionality of 40,578. Thus, our form vectors are extremely sparse. Defining vector sparsity as the ratio of zero-valued elements to the total number of elements in the vector, our form vectors have a sparsity equal to 0.99 ($SD = 0.009$).

Thus, for both NDL and LDL, we have two matrices, a 131,372 $\times$ 40,578 form matrix $\boldsymbol{C}$, and a semantic matrix $\boldsymbol{S}$ which is of dimension 131,372 $\times$ 4741 for NDL and of dimension 131,372 $\times$ 4609 for LDL, irrespective of whether or not learning is incremental. For non-incremental learning, the weight matrix (or matrix of coefficients)

Table 2. The four models considered in the simulations.

	Outcomes	
Training Method	lexomes	semantic vectors
gradient descent	Rescorla-Wagner (NDL)	Widrow-Hoff (LDL)
matrix inversion	NDL classifier (Danks, 2003)	LDL multivariate multiple regression

is obtained by solving the system of equations defined by $\boldsymbol{C}$ and $\boldsymbol{S}$ as explained in the preceding section. For incremental learning, for each of n iterations through the data, learning proceeds step by step through the learning events defined by the rows of the matrices.

5.2. Results

Our first simulation experiment takes as starting point the NDL model of Shafaei-Bajestan and Baayen (2018), which maps the (discrete) auditory cues onto lexomic semantic vectors. As this study also considered only monomorphemic words, the task given to the model is a straightforward classification task. Figure 3 presents the classification accuracy on the training data at the endstate of learning by means of a horizontal dashed line. The solid line presents model accuracy when the Rescorla-Wagner learning rule is used. The first data point represents accuracy after one iteration, the value reported by Shafaei-Bajestan and Baayen (2018). Subsequent datapoints represent accuracy after 100, 200, ..., 1000 iterations through the dataset. Importantly, the endstate accuracy emerges as the asymptote of incremental learning. Apparently, it is not the case that there is a sweet spot at which incremental learning should be terminated in order to avoid overfitting.

In the second simulation experiment, we replaced one-hot encoding of semantic vectors with the distributional vectors of the TASA1 semantic space. Figure 4 illustrates that training with the Widrow-Hoff learning rule to discriminate between words' semantic vectors also slowly moves towards the endstate asymptote. However, overall accuracy of this model is substantially reduced to only 33.7% at equilibrium. Although again incremental learning with the Widrow-Hoff learning rule is confirmed to be incremental regression, with estimated coefficients asymptoting to those of a multivariate multiple regression analysis, the drop in accuracy is unfortunate. Why would it be that moving from one-hot encoded semantic vectors to distributional vectors is so detrimental to model performance?

A possible reason is that the classification problem with one-hot encoded vectors is easier. After all, one-hot encoded vectors are all completely orthogonal, which should make them better discriminable. With semantic vectors, words will be more similar to other words, and this is likely to have rendered them more difficult to learn. To test this explanation, we replaced the TASA1 vectors by random vectors sampled from a uniform distribution over $[0, 1)$.The resulting pattern of learning is virtually identical to that shown in Figure 3 (figure not shown). Thus, as long as semantic vectors are orthogonal, incremental learning with Rescorla-Wagner and with Widrow-Hoff produces exactly the same results.

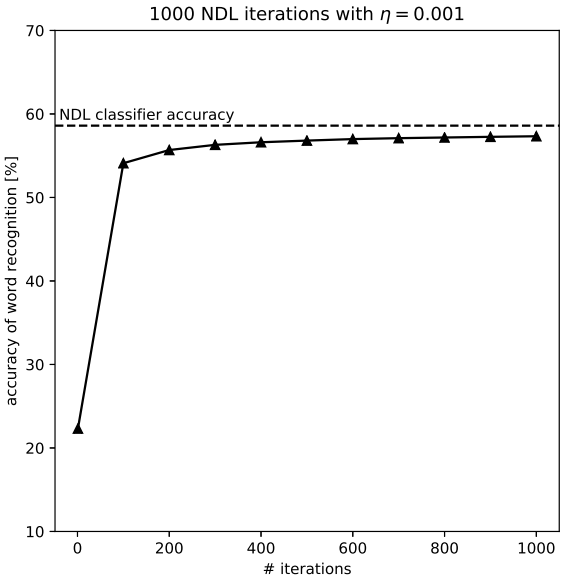

Figure 3. NDL learning curve, using one-hot encoded semantic vectors. NDL accuracy using the Rescorla-Wagner learning rule approaches the asymptotic equilibrium state of the NDL classifier.

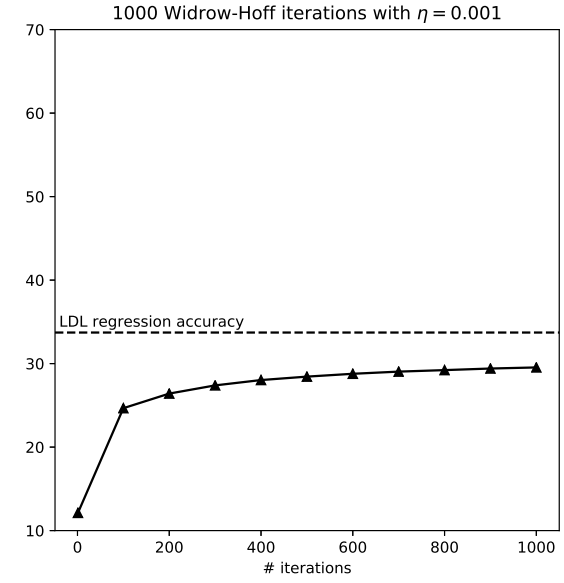

Figure 4. Widrow-Hoff learning curve, using semantic vectors derived from TASA. Widrow-Hoff accuracy approaches the asymptotic state approximated by an LDL model, but accuracy is substantially reduced compared to Figure 3.

5.3. Discussion

The three simulation experiments all show that there is no sweet spot for incremental learning, when accuracy is evaluated on the training data. The endstate is the theoretical asymptote for learning when the number of epochs n through the training data goes to infinity. Below, in section 8, we address the question of how evaluation of accuracy under cross-validation develops over training epochs. Our simulations also show that the Widrow-Hoff and Rescorla-Wagner learning rules produce identical results, as expected given the mathematics of these learning rules. Furthermore, our simulations clarify that model performance, irrespective of estimation method, critically depends on the orthogonality of the semantic vectors. In section 7, we return to this issue and we will present a way in which similarity (required for empirical linguistic reasons) and orthogonality (required for modeling) can be properly balanced. The next section, however, first addresses the question of whether the features extracted from the auditory input can be further improved.

6. Learning with enhanced auditory features

Thus far, we have used the discrete acoustic features proposed by Arnold et al. (2017), named frequency band summary features (FBSF). These features log patterns of change in the power spectrum over time at 21 frequency bands that represent the perceptual scale of the pitch. These patterns of change are extracted for stretches of speech bounded by minima in the smoothed Hilbert envelope of the speech signal's amplitude (henceforth, chunks), and summarized using a pre-defined set of descriptive statistics[7]. The number of different FBSFs for a word are a multiple of 21 and the total number of features typically ranges between 21 and 84, depending on the number of chunks.

The FBSFs are inspired by the properties and functions of the cochlea, and the basilar membrane in particular. FBSFs decompose the incoming continuous signal in the time domain into a sum of simple harmonics through a Fast Fourier Transform, similar to the basilar membrane's response to a complex sound with multiple excited regions corresponding to the constituent frequencies, which is enabled by its tonotopic organization. Furthermore, power values present at different frequencies are summed over a series of filters obtained according to the MEL formula presented in Fant (1973), similar to the cochlea's separation of the input signal into overlapping auditory critical filter banks that jointly are responsible for the nonlinear relationship between pitch perception and frequency. In addition, power values are then log transformed, similar to the logarithmic relationship between loudness perception and intensity. Finally, the algorithm summarizes change patterns over time in the power values at different frequency bands by means of discrete features. An FBSF extracted from the acoustic input is assumed to correspond to, at a functional level, a cell assembly that is sensitive to a particular pattern of change picked up at the basilar membrane and transferred in the ascending auditory pathway to the auditory cortex.

This approach to signal processing differs from standard approaches, in that the focus is on horizontal slices of the frequency spectrum, corresponding to different bands on the basilar membrane, instead of the vertical slices in the spectrum that correspond to phones. Although the results obtained with this approach are promising and compare favorably with ASR techniques from deep learning when tested on isolated word

[7]The complete set contains the frequency band number, the chunk number, the first, the last, the minimum, the maximum, and the median of the power values.

recognition (see Arnold et al., 2017; Shafaei-Bajestan and Baayen, 2018, for detailed discussion), one problem with FBSFs is, however, that their temporal resolution is restricted to time intervals that are of the order of magnitude of the time between minima in the Hilbert envelope, which correspond roughly to syllable-like units. As a consequence, the model has insufficient temporal granularity to be able to model cohort effects. Furthermore, the discretization of patterns of change in spectral frequency bands, necessitated by the use of the Rescorla-Wagner learning rule within the framework of NDL, may come with a loss of precision. We therefore investigated whether, within the framework of LDL, this approach can be enhanced. In what follows, we define new features, Continuous Frequency Band Summary Features (C-FBSF), and we will show that they have better performance than their discrete counterparts.

6.1. Method

Pseudo-code for C-FBSF extraction is given by algorithm 1 (displayed below) that takes the audio file of a word as input and returns a feature vector for the word by concatenation of feature vectors for word's chunks. To assemble a feature vector for a chunk, algorithm 1 finds the chunking boundaries defined by extrema (minima or maxima) in the Hilbert envelope using algorithm 2 and calls algorithms 3 and 4 on each chunk. Algorithm 3 performs a spectral analysis on a chunk and returns the logarithm of power values at MEL-scaled frequency bands. Algorithm 4 summarizes the power spectrum of a chunk and returns a feature vector for the chunk. Summarization of a chunk's spectral information can be attempted in various ways. In the present implementation, from the sequence of log power values at a particular frequency band and a particular chunk, we extract 1) frequency band number, 2) an order-preserving random sample of length 20, and 3) correlation coefficients of the values at the current frequency band with those of the following bands. In this way, we do not use a predefined set of descriptive statistics. In addition, the correlation structure between the frequency bands of a chunk is made available for learning. For a given chunk, all of the possible cross-frequency-band correlation values are represented in the feature space, with the order of dimensions set rather arbitrarily. We obtain a 651-dimensional vector of real numbers for each chunk. All feature vectors for words are padded with trailing zeros to match the length of the feature vector for the word with the largest number of chunks, and have a dimensionality of 6510.

The C-FBSF algorithm identifies chunk boundaries in a word at the maxima of the signal's envelop. Results are similar, but slightly inferior, when maxima are replaced by minima. The top and middle panels in figure 10 present the chunking boundaries for the audio signal of the word *captain* in the waveform and in the power spectrum, respectively. The python implementation of the algorithm, available in the python package **pyLDLauris**, allows the user to fine-tune the chunking criteria.

The audio tokens in our data set are, on average, split into 2.23 chunks (N =131,372, SD = 1.07, range: $1 - 10$) by the C-FBSF algorithm. There is a strong positive correlation between the duration of words and the number of chunks detected by the C-FBSF algorithm, $r(131372) = 0.85$, $p < 0.001$. The average chunk duration is 114 ms (N =292,776, SD = 0.06, range: $10 - 561$). Figure 5 clarifies that the duration of chunks is log-normally distributed.

We built two models, one using the FBSF features of Arnold et al. (2017), the other using the new C-FBSF features. Word meanings were represented by means of semantic vectors from the vector space extracted from the TASA corpus with 23,561

Algorithm 1 Steps for C-FBSF extraction

function GETCFBSF($file$)
  wav, sr ← read wave data and sampling rate from audio file $file$
  if $sr \neq 16000$ **then** resample wav to 16000
  end if
  chunks cs ← GETCHUNKS(wav)
  word's vector wv ← empty vector
  for all $chunk \in cs$ **do**
    chunk's power spectrum ps ← GETLOGMELPOWSPEC($chunk$)
    chunk's vector v ← GETSUMMARY(ps)
    expand wv with v
  end for
  return wv
end function

Algorithm 2 Steps for chunking a stretch of audio

function GETCHUNKS(wav)
  analytic signal a ← Hilbert transform of wav
  envelope e ← modulus of the complex-valued a
  window w ← a boxcar window
  smoothed window se ← the convolution of e and w
  indices i ← arguments of the maxima for e
  chunks c ← segments of wav split by i
  return c
end function

Algorithm 3 Steps for spectral analysis of a stretch of audio

function GETLOGMELPOWSPEC(wav)
  spectrogram sg ← fast Fourier transform of wav using non-overlapping 5 ms windows and 512 channels
  power spectrum ps ← modulus of the complex-valued sg, squared
  filterbank fb ← 21 auditory critical bands computed based on the MEL formula of O'shaughnessy (1987)
  mel power specturm mps ← sum power values in ps in each filter of fb
  log mel power spectrum $lmps$ ← log(mps)
  return $lmps$
end function

Algorithm 4 Steps for summarizing the changes in spectral information into a vector

function GETSUMMARY(ps)
  v ← empty vector
  for $i = 1 \rightarrow 21$ **do**
    append i to v
    append an order-preserving random sample of 20 from $ps[i]$ to v
    for $j = i + 1 \rightarrow 21$ **do**
      append correlation between $ps[i]$ and $ps[j]$ to v
    end for
  end for
  return v
end function

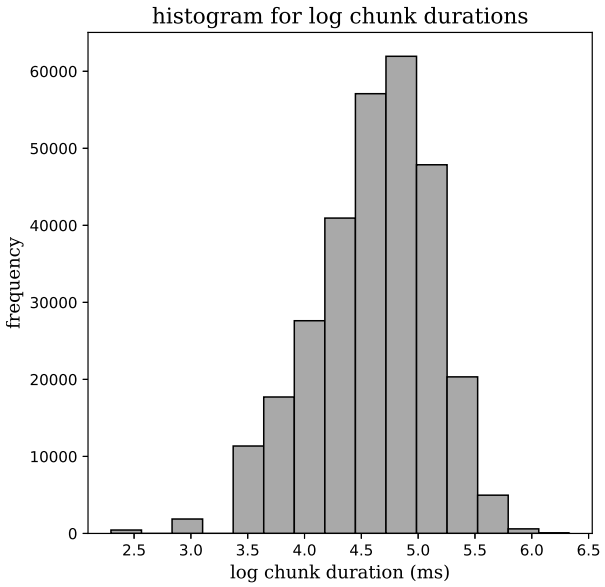

Figure 5. Distribution of chunk durations in C-FBSF. Chunk duration follows a lognormal distribution.

Table 3. Comparison of the performance of LDL using FBSF and C-FBSF (accuracy of correct word recognition [%]).

Feature extraction	Training Accuracy[a]	Testing Accuracy[b]
FBSF	38.9	6.9
C-FBSF	16.2	11.3

[a] Recognition accuracy on training data.

[b] Mean recognition accuracy on test data over 10 cross-validation folds.

word types constructed by Baayen et al. (2019). Henceforth, we denote this space as TASA2. Vectors from TASA2 are 4609 long. TASA2 contains vectors for 4377 of the total number of 4741 word types in the data set.

6.2. Results

In order to evaluate the accuracy of the C-FBSF features, we evaluated recognition accuracy both on the training data itself, and on held-out data using cross-validation. By comparing the accuracy values, we gain further insight into the extent to which models are overfitting the training data. Table 3 presents accuracy in percentage correct for LDL in recognizing word tokens of the data set for which a TASA2 vector is available (N =123,719). When LDL is provided with the sparse binary vectors of FBSF, it learns the training data well (accuracy 38.9%), but accuracy under cross-validation plummets to 6.9%, a clear warning that the model is overfitting. When LDL is supplied with C-FBSF, its performance on the training data is less, compared to the original features, at 16.2%, but performance under cross-validation reduces to only 11.3%, nearly double the performance of the original features, and substantially outperforming the deep learning network tested by Shafaei-Bajestan and Baayen (2018).

What do the new acoustic features represent, and how should they be interpreted?

Table 4. Frequencies for chunks comprising one phone used in t-SNE analysis.

Phone in Chunk	Chunk Frequency
/b/	2588
/p/	1985
/d/	6603
/t/	11633
/a/	2114
/e/	2600
/i/	6866

Questions such as these are not straightforward to answer for the discrete FBSF features. Since the new C-FBSF features are continuous rather than discrete, some insight into what they represent can be obtained relatively straightforwardly by means of clustering methods. Following Baayen et al. (2018), we reasoned that if our acoustic features are understood as the functional equivalent of cell ensembles monitoring for patterns of change in cochlear frequency bands, then the question arises of how such ensembles might be organized in a two-dimensional plane, where this plane is a very rough approximation of some area of the cortex. Given that some topographical clustering of phones has been observed in medical studies (see Cibelli et al., 2015, and references cited there), one may expect phone-like clustering when C-FBSF features, which are high-dimensional vectors, are projected onto a two-dimensional space. We used the t-SNE clustering algorithm (Maaten and Hinton, 2008) as implemented in the `scikit-learn` Python library (Pedregosa et al., 2011) to perform the clustering. t-SNE is a non-linear technique that is particularly well suited for the visualization of high-dimensional data and that is often used for interpreting patterns of activation in deep learning models.

To obtain a two-dimensional representation of the C-FBSF features, we proceeded as follows. As a first step we extracted, for all the chunks, its 651-dimensional C-FBSF feature vector together with a list of the phones present in that chunk. We computed the list of phones in a chunk by adding the phone boundaries provided by the aligner to the chunk boundaries produced by the C-FBSF chunking algorithm. A phone is categorized as contained in a chunk if all or most of its audio signal is present in the chunk. This resulted in a matrix with for each phone a 651-dimensional row vector of the C-FBSF feature for the chunk in which that phone was present. This matrix was then first subjected to a Principal Components Analysis, resulting in an orthogonalized space that in a final step was presented as input for the t-SNE.

In what follows, we zoomed in on those chunks that contained one of the phones /b,p,d,t,a,e,i/ and that did not fully contain any other phones. Table 4 reports the frequency of occurrence for all pertinent chunks-phone combinations. Figure 6 presents the locations in the t-SNE topographic map of the chunk-phone combinations for /b/ and /d/ (left panel) and /p/ and /t/ (right panel). For both pairs of consonants, we see some clustering with fractal-like properties (Mandelbrot, 1982). The center-most clusters of points predominantly represent /d/ and /t/ respectively, with a subcluster of /b/ and /p/ in their respective peripheries. This pattern repeats itself in the smaller satellite clusters. Comparing the two plots, it is noteworthy that /d/ (blue) and /t/ (purple) show highly similar clusters, perhaps unsurprisingly given that they only differ in voice onset time. The isomorphism between /b/ and /p/ is less clear. This, however, may be due to the substantially smaller number of data points present for these phones. Overall, the similitude of the two plots shows that the labial-alveolar contrast is in like

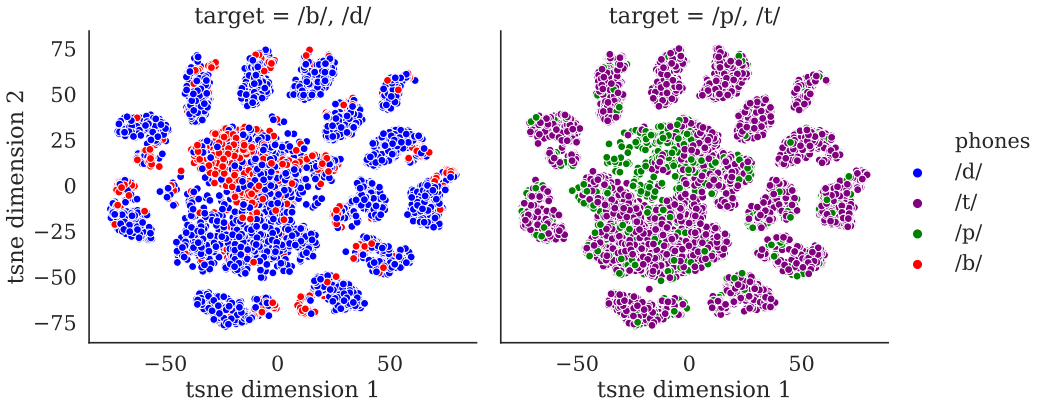

Figure 6. topographic map of stop consonants visualized by t-SNE clustering.

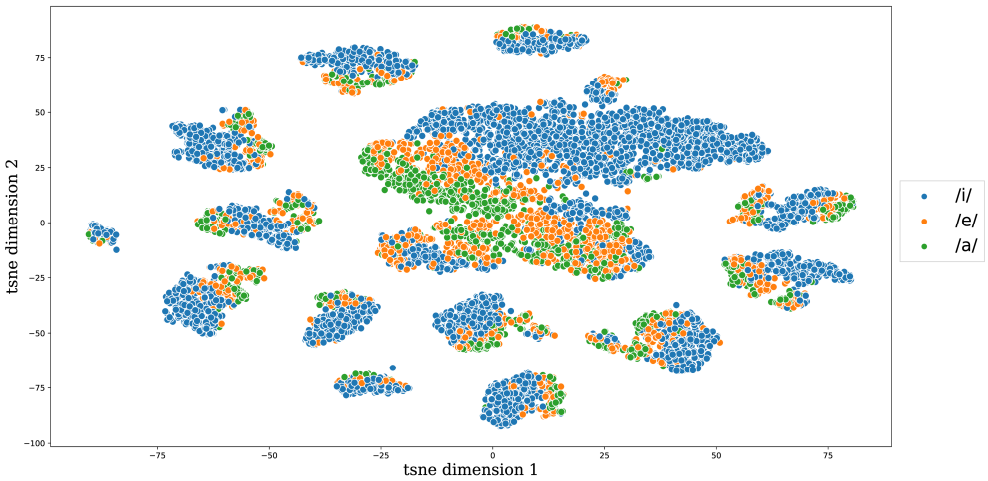

Figure 7. topographic map of peripheral vowels visualized by t-SNE clustering.

manner captured for both /b/-/d/ and /p/-/t/. A similar fractal-like structure emerges for the vowels /a, e, i/, as shown in Figure 7, with recurring leaky separation of /i/ from /e/ and /a/, and some further separation within the /e/ and /a/ clusters. The difference between consonants in Figure 6 and vowels in Figure 7 is naturally brought out by the features.

6.3. Discussion

From the LDL simulations using FBSF and C-FBSF, we conclude that the features from C-FBSF substantially attenuate the over-fitting problem that characterizes the LDL model when using the FBSF features. For generalization, working with real-valued acoustic features instead of discrete summary features offers a clear improvement in performance. We therefore use C-FBSF in the simulation experiments presented in section 8.

From the t-SNE clustering analysis of C-FBSF features we can conclude that, even though these features slice the spectrogram horizontally, along cochlear frequency bands instead of vertically, phone by phone, they nevertheless preserve substantial information about phone classes. At the same time, the overlap between the consonant maps and the vowel map indicates that phones need not be uniquely represented in the map, but will often share a position in the map with other phones. This makes perfect sense from a phonetic perspective, as co-articulation is ubiquitous. For the present phones, for instance, place of articulation of the stops is signalled by the formant transitions in the vowels they co-occur with. Importantly, even though in our model the theoretical construct of the phoneme does not play a role, the C-FBSF features are sufficiently rich to capture similarities and dissimilarities between phonemes. These similarities, in turn, co-determine the mapping from form onto meaning. Thus, in our approach, phones are not emergent on some hidden layer of a deep learning network, but rather are implicit in the input vectors.

In the next section, we consider whether the representation of meaning in NDL and LDL can be enhanced further.

7. Learning with enhanced semantic vectors

In section 5, we observed that semantic vectors derived from the TASA corpus underperformed considerably compared to either one-hot encoded semantic vectors or near-orthogonal vectors of random numbers. This result suggests that ideally semantic vectors should strike a balance between being well discriminable (close to orthogonal) while at the same time reflecting the semantic similarities that native speakers perceive when judging word pairs (see, e.g., the MEN dataset compiled by Bruni et al., 2014).

Would semantic vectors as constructed by means of machine learning methods in the computational linguistics community (such as `word2vec`, see `https://code.google.com/archive/p/word2vec/` and Mikolov et al. (2013)) provide a proper balance? Although these vectors are very good predictors of human-percieved semantic similarity ($r(1176) = 0.76, p < 0.001$ for the MEN dataset), they are trained on approximately 100 billion words from the Google News^{TM} data. This volume is far more than anyone will ever encounter in their lifetime. To put that in perspective, imagine hearing 1000 words per hour for every hour of every day of 80 years of life. Under these extreme circumstances, one would only have heard some 700 million words. Thus, from a cognitive perspective, such vectors are unrealistic, as they are tuned to vastly more knowledge (including the full wikipedia) than anybody can ever assemble in a lifetime. But training with massive data may give rise to semantic vectors that are both distinct enough to be both discriminable and faithful to semantic similarity.

In what follows, we consider whether semantic vectors trained on 'only' 10 million words taken from the TASA corpus can be enhanced by adding some small amount of random noise. Technically, the idea is that by adding some noise, the semantic vectors become more discriminable, and therefore can be better predicted from the acoustic feature vectors. Conceptually, the idea is that the noise represents, however crudely, the way in which words' semantics are affected by sensory information (vision, hearing, olfaction, and touch) over and above what is captured by textual co-occurrence patterns.

7.1. Method

We contrasted learning with four semantic spaces. The first two are the semantic spaces TASA1 and TASA2 that we introduced in previous sections. Two additional vector spaces were built by element-wise addition of a noise vector $\boldsymbol{n}$ to the semantic vectors of TASA2. Noise vectors were sampled from a Gaussian distribution with $\mu = 0$ and standard deviations 0.001 and 1.0 respectively.

We assessed the degree of orthogonality of the resulting four semantic spaces with two evaluation metrics, the average correlation and the average variance. We computed the average correlation for a semantic space by taking the average of the Pearson r correlation coefficients for all pairs of semantic vectors. The lower the average correlation, the more orthogonal the vector space is. The average variance for a semantic space is the average of the variances for all semantic vectors in the space. The higher the average variance, the more orthogonal the vector space is likely to be.

The data to which we applied these measures comprised all 4377 word types for which semantic vectors are available in TASA and which appear in our speech dataset. LDL models were trained to discriminate distributional features of the different TASA semantic spaces using FBSF features. Model accuracy was evaluated on the training set. The extent to which a semantic space captures the semantic structure of the lexical representations was examined on the MEN database (Bruni et al., 2014) that provides for 3000 word pairs crowd sourced ratings of semantic similarity. For 1176 word pairs, semantic vectors are available in all semantic spaces for both words. For this subset of words, we evaluated to what extent our semantic vectors matched human-perceived similarity.

7.2. Results

Figure 8 plots the training accuracy of the LDL model against average correlation of the semantic space in the left panel, and against the average variance of the semantic vectors in the right panel. LDL training accuracy increases with lower average correlation and higher average variance, as expected. TASA1 vectors obtained from a smaller subset of TASA have the least discriminated features and are not well discriminated by LDL. More data in training of TASA2 compared to the training of TASA1, resulted in a vector space with more distinct vectors, thereby facilitating learning. Addition of a tiny amount of noise boosted accuracy substantially. Increasing the standard deviation of the noise a thousandfold offered only a minor further improvement.

Table 5 lists the Pearson's coefficients for the correlation between the MEN ratings for pairs of words and the semantic similarities of the corresponding two semantic vectors from our semantic spaces. The gain in capturing semantic similarities of words achieved in TASA2 compared to TASA1 is likely due to a larger subset being used when training the TASA2 space. Addition of a tiny amount of Gaussian noise brought down the correlation somewhat while at the same time, as demonstrated above, affording a substantial boost in prediction accuracy. Addition of substantial noise almost completely removed lexical similarity structure from the vectors, while offering only a modest additional accuracy gain.

7.3. Discussion

Addition of a tiny amount of noise to the TASA2 vectors boosted accuracy, evaluated on the training data, by about 15% to 53.8%. When we in addition consider accuracy

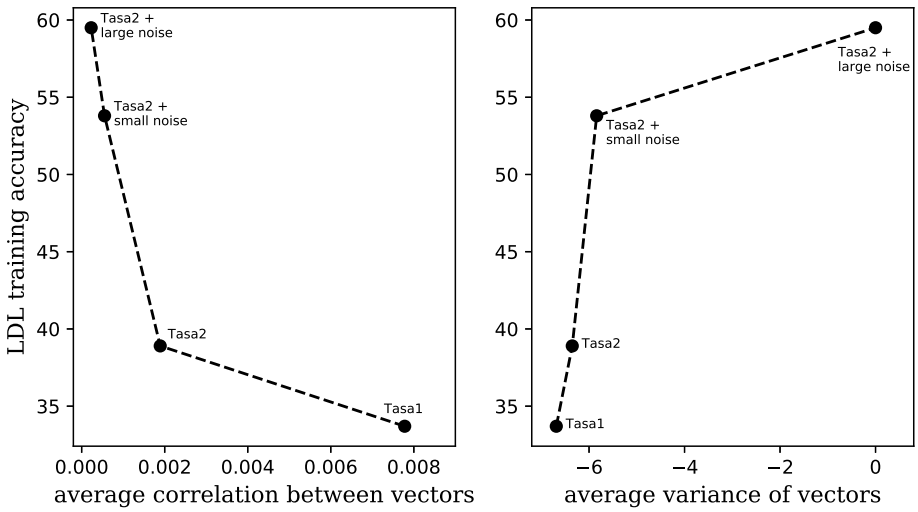

Figure 8. Orthogonality measures as predictors of LDL accuracy evaluated on the training data.

Table 5. Similarity structure captured by the four types of semantic vectors.

Vector space	Correlation with MEN similarity ratings	p-value
TASA1	0.24	< 0.0001
TASA2	0.59	< 0.0001
TASA2 plus small amount of noise	0.47	0.4946
TASA2 plus large amount of noise	0.02	< 0.0001

for these vectors under 10-fold cross-validation, we also observe an improvement from 6.9% to 11.1%. Interestingly, LDL performance with `word2vec` vectors was not as good (52.6% accuracy on training data, but only 8.5% averaged on 10 folds of test data). We therefore use the TASA2 vector space with a tiny amount of noise added in our final simulations presented in the next section, which introduces our best and definitive model.

8. Putting it all together

Our final simulation study combines the insights of the preceding sections to define an improved discriminative model for auditory word recognition that we have named LDL-AURIS. This model makes use of C-FBSF to represent words' auditory forms, it uses empirical semantic vectors derived from TASA with a small amount of noise added, and it estimates network weights using either multivariate multiple regression or iterative application of the Widrow-Hoff learning rule.

In what follows, we report on the model's performance, focusing on two main questions. First, the accuracy of the new model is of interest, both for the training data on the one hand, and under 10-fold cross-validation on the other hand. Second, do the C-FBSF features have better temporal granularity than the FBSF features, and if so, do they make it possible to now predict cohort effects?

Table 6. Summary of GAM model

A. parametric coefficients	Estimate	Std. Error	t-value	p-value
(Intercept)	-1.6120	0.0100	-161.2772	< 0.0001
B. smooth terms	**edf**	**Ref.df**	**F-value**	**p-value**
s(logdur)	3.9816	3.9998	15377.9701	< 0.0001
s(logfreq)	3.9902	3.9999	20071.5283	< 0.0001

8.1. Accuracy

LDL accuracy was 25% on training data and average LDL accuracy under 10-fold cross-validation was 16%. Compared to the model presented in Baayen et al. (2019), the model showed an 8% decrease in training accuracy but an 8% increase in test accuracy, considerably reducing the extent of the over-fitting problem. When we consider the number of target semantic vectors among the top 5 and top 10 words showing the strongest correlations with the predicted semantic vector, accuracy increases to 57% and 75% on training data and to 37% and 50% on test data. Thus, model accuracy (under cross-validation) comes close to the lower bound of the range of human recognition accuracy documented for single word recognition tasks (Arnold et al., 2017; Pickett and Pollack, 1963; Pollack and Pickett, 1963). The performance of our model contrasts favorably with the recognition rate of Mozilla Deep Speech, which was roughly 10% lower (Shafaei-Bajestan and Baayen, 2018). It is also noteworthy that the data on which we train and test the model comes from many different speakers from a wide range of backgrounds, and that human listeners typically require some time to adjust to the speech of unfamiliar speakers (see also Wieling et al., 2014).

Human auditory word recognition is sensitive to word frequency (see, e.g. Connine et al., 1993; Seidenberg and McClelland, 1989), with higher frequency words being recognized more accurately. We used a generalized additive model with a logistic link function, using the **mgcv** package for R (Wood, 2017), to predict whether the model correctly identified a word token, using log word frequency and log duration as predictors. Partial effects are shown in Figure 9 and Table 6 provides the model summary. Longer words are recognized more often by the model, and the same holds for more frequent words. The advantage for longer words, given the negative correlation of frequency and length, shows that the model does not depend on only frequent use, but is also properly sensitive to the amount of information in the speech signal. The rightmost panel of Figure 9 shows the frequency effect predicted for auditory lexical decision. These predictions are defined as the reciprocal of the probability of correct recognition produced by the regression model. The nonlinear effect of frequency, with a leveling off for higher frequencies, resembles the nonlinear effect typically observed in reaction time studies of reading (see, e.g. Baayen, 2005; Ramscar et al., 2014). A similar pattern also characterizes the auditory lexical decision times in the MALD database (Tucker et al., 2017) (model not shown). Thus, qualitatively, the model provides a good approximation of the shape of the word frequency effect.

8.2. Cohort effects

The FBSF features developed by Arnold et al. (2017) cover stretches of speech that are syllable-like. Their temporal granularity is too coarse to allow modeling of cohort effects in auditory word recognition. The C-FBSF features of our new model cover shorter stretches of speech. This makes it possible to use the trained model to assess how model predictions develop as the speech signal unfolds over time. Specifically, we

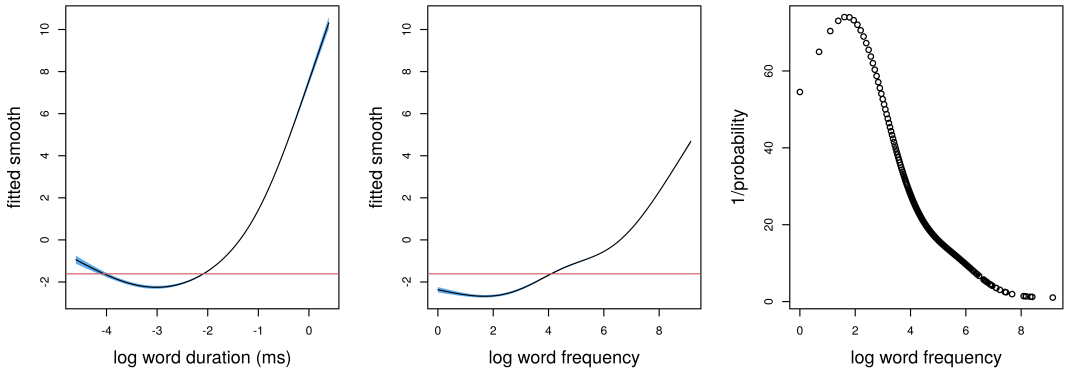

Figure 9. Partial effects of log word frequency (left) and log duration (center) on model accuracy according to a logistic generalized additive model. The right panel presents predicted auditory lexical decision latencies, derived from the predicted error probabilities.

consider how well the model's predicted semantic vector for a given acoustic input approximates the targeted semantic vector at each point in time at which a chunk has been identified and the corresponding C-FBSF have been extracted. At the point in time where the first chunk has been identified, the input vector contains just the features of this first chunk. All remaining values in the form vector are set to zero. At the next point in time, the features extracted from the second chunk are added. This process is repeated until the after identification of the last chunk, the final set of features is added. At this final point in time, the model predicts exactly the same semantic vector as when all features are made available to the model simultaneously. In other words, the chunks define a path in semantic space that necessarily ends at the point in semantic space that is predicted when all features are available to the model. For the conceptualization of understanding as a path in lexical space, see also Elman (2009).

The lower panel of Figure 10 illustrates the time course of lexical processing for the target word *captain*, comparing the correlation of the predicted semantic vector with that of *captain* for three input words, *cap*, *capital*, and *captain*. There are 8, 22, and 7 audio tokens respectively for these three words in our dataset. From this list of tokens, we randomly selected one audio token for each word, with as constraint that, for plotting reasons, the number of chunks in the competitors do not exceed the number of chunks in the target word. The upper panel of Figure 10 presents the audio signal of the selected token for *captain*, together with its Hilbert envelope. The red vertical lines highlight where the Hilbert envelope of the token *captain* reaches a local maximum. The selected audio tokens of *cap* and *capital* have their local maxima at 0.17 s and 0.14 s, and have a total duration of 0.37 s and 0.41 s, respectively. The center panel presents the MEL spectrogram corresponding to the audio token of *captain* shown in the top panel, with on the vertical axis the 21 auditory filter banks inspired by the tonotopy of the basilar membrane in the cochlea .

With respect to the time course of lexical processing, in the beginning, at $t = 0$, no auditory information is present and the feature vectors for all words are filled with 0s only. On the presentation of the first chunk, the model has detected that all three candidates approximate the target semantic vector to some extent, with *captain* already taking the lead. Since the word *cap* has most of the signal for /æ/ in its second chunk,

it is not a strong competitor at the first chunk. The average correlation value for all the 55 tokens of 9 types that start with /kæp/ is 0.022 at the end of their first chunks. All seven occurrences of *captain* are correlated with the target vector well above the mean at the end of the first chunk ($M = 0.041$, $SD = 0.009$). By the end of chunk two, *cap* and *capital* die out and lose out further to *captain*. Occurrence of /n/ in the final chunk pushes the predicted semantic vector of *captain* even closer to the target vector.

In classical models of lexical processing, in which words' forms are accessed in parallel, the present time course plot would be understood as demonstrating multiple access and multiple simultaneous assessment unfolding efficiently in real time. Marslen-Wilson (1987) argued that a model for spoken word recognition should meet three functional requirements. First, models should properly reflect 'multiple access'. Our model can be interpreted to indicate that all three words are accessed simultaneously, forming a class of potential word candidates compatible with the sensory input. Second, models should reflect 'multiple assessment' of word candidates. Our model appears to meet this requirement as by the end of the first chunk, where multiple candidates are compatible with the input, the system already ranks candidates. Third, models should accommodate multiple access and assessment in real time. Our model also satisfies this requirement: Within about 200 ms from word-onset, the target word is beginning to be recognized, and as more chunks become available, candidates' ranking is recalibrated.

However, the general conceptualization underlying our model differs from that of Marslen-Wilson (1987). Our model construes understanding as it develops over time as speech comes in as a path through lexical space (cf. Elman, 2009). There are no discrete processing stages nor a final state in which a word has been accessed, but rather a gradual process of uncertainty reduction (see also Baayen et al., 2015; Ramscar, 2013; Ramscar et al., 2013) that, importantly, does not need to resolve into a one-winner-take-all state of absolute certainty. Thus, even though the conceptualization of the process of "lexical access" is different from that of TRACE or Shortlist, our model does show the kind of temporal dynamics that has been an important explanandum for classical models.

9. General Discussion

The computational model for auditory word recognition laid out in this study builds on an earlier model proposed by Arnold et al. (2017), enhancing it in several ways. First, real-valued feature vectors extracted from the speech signal replace discrete binary vectors, while maintaining the important insight that cochlear frequency bands should inform feature engineering for cognitive modeling. Second, discrete binary vectors with one-hot encoding for words' meanings are replaced by real-valued semantic vectors. By adding a small amount of noise to vectors derived from a relatively small corpus (TASA) using distributional semantics, semantic vectors are obtained that are sufficiently discriminable while respecting semantic similarities between words. Third, instead of using incremental learning using the Rescorla-Wagner learning rule, with one pass through the data, we estimated the endstate of learning using the mathematics of multivariate multiple regression, which simulation studies show to offer greater accuracy.

Together, these new design features offer the following advantages. First, overfitting on the training data is substantially reduced, whereas prediction for unseen data,

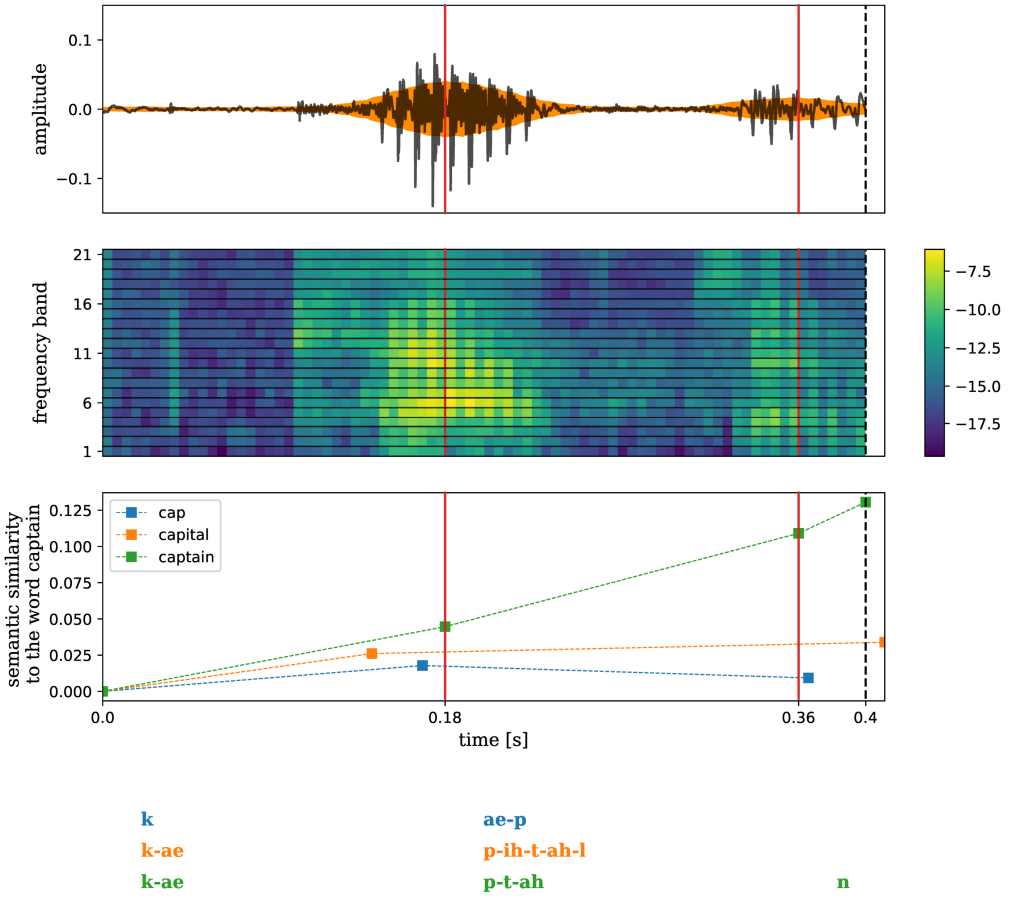

Figure 10. Time course of lexical access for the target word *captain*. Top panel: the waveform for a token of the word *captain* in dark gray, and the Hilbert amplitude envelope of the signal in orange. The red vertical lines indicate chunk boundaries, located at the arguments of local maxima of the Hilbert envelope at 0.18 s and 0.36 s. The dashed black vertical line at 0.4 s indicates the end of word boundary. Mid panel: the corresponding log MEL spectrogram split at 21 auditory filter banks shown on the y-axis. Lower panel: semantic similarity, as measured by Pearson's correlation coefficient r, to the target word as a function of time, for the target word *captain* and two competitors *cap* and *capital*. The transcripts at the bottom show the phonemes present in the chunks for the three words.

evaluated by means of 10-fold cross-validation, improved substantially. Second, the new acoustic features provide enhanced temporal granularity, allowing the model to correctly reflect cohort-like effects as the speech signal unfolds over time. Third, the new acoustic features are better interpretable, as shown by projecting form vectors onto a two-dimensional plane with t-SNE. In this plane, phone-like clusters emerge. This shows that even though our acoustic features are based on horizontal slices from the spectrogram (following cochlear frequency band separation) instead of on vertical slices representing phones, information about phones is retained. The phone-like topological clustering that emerges from the t-SNE clustering analysis may shed light on the topological clustering observed for phones in the cortex (Cibelli et al., 2015). The neuroanatomical evidence would at first sight indicate that phones are physiologically real. However, under our interpretation, such phone clusters emerge because they allow for efficient localization of acoustic features in a way that we anticipate will turn out to be highly energy-efficient, reducing the metabolic costs of lexical processing.

In this study, we considered monomorphemic words, and derived words with monomorphemic base words, but no inflected variants of these words. However, the general framework within which the present study is conceived, the 'discriminative lexicon' as outlined in Baayen et al. (2019), sets up semantic vectors for inflected words by taking the semantic vector of the base word and adding the semantic vector of the pertinent inflectional function. Shafaei-Bajestan and Baayen (2020) show that an auditory model along the lines outlined in the present study, but that includes also inflected words, performs well, also for inflected forms that are not in the training data but that are variants of known basewords. In other words, the model is productive for novel inflected forms. This property sets the present model apart from 'full-listing' models of auditory comprehension such as TRACE, Shortlist-B, and Diana.

Model performance was evaluated on real spontaneous speech, from many different speakers, taken from the NewsScape corpus of the Distributed Little Red Hen Lab. Speaker normalization was not necessary, and there were no hyperparameters to tune. The only free parameter of the model is the learning rate, which we set to 0.001 as in previous studies, and never changed. The changes that we implemented for our new model all concerned the vectorial representations that define words' forms and meanings. Given the matrices that define the form space and the semantic space of the mental lexicon, the mathematically well-understood mathematics of multivariate multiple regression are all that is required. We think, but have not yet been able to provide proof, that the high dimensionality of our form and semantic vectors, combined with high sparsity, are prerequisites for multivariate multiple regression to be successful. We are beginning to find that when high-dimensional vectors are projected and compactified into low-dimensional spaces, deep learning networks are essential. For linguistic analysis, our 'wide-learning' approach offers two advantages over deep learning. First, it offers interpretable machine learning. Representations of form and of meaning are linguistically motivated. For instance, our semantic vectors straightforwardly quantify the collocational strengths of a given word with a large number of other words. As we have shown above, our form vectors respect cochlear separation and yet retain rich information that drives phone-like clustering in two dimensions. Second, results do not depend on a large number of hyper-parameters — for training, we fall back on the well-understood mathematics of regression.

Human learning is incremental, and it is therefore important that the weights of the regression model can be learned incrementally with the Widrow-Hoff learning rule, which defines incremental regression. In this study, we have seen that incremental learning requires many passes through the data. This is unrealistic for human learning,

as we never experience the same sequence of learning events repeatedly. We are exposed to ever more of rather similar input, but never exactly the same input. To some extent, our incremental simulations approximate this ever changing input by randomizing the order of learning events for each pass through the data. But reusing the same audio files remains unrealistic. As a consequence, the accuracy of the endstate model is also conditional on the training data, which emphasizes the importance of working with large and varied data, and not with synthesized or laboratory speech. Fortunately, the amount of audio offered by the Distributed Little Red Hen Lab is so huge that we can train the model incrementally on thousands of hours of audio, without ever having to repeatedly present a specific acoustic word twice. An important goal for future research is to clarify how well our new model performs when challenged with such large volumes of speech.

Another important goal for the present research programme is to move from modeling isolated word recognition to the modeling of the understanding of continuous speech, perhaps along the lines of Baayen et al. (2016b). Our model outperforms on isolated word recognition two deep learning models that we have explored (see Arnold et al., 2017; Shafaei-Bajestan and Baayen, 2018, for further details). As isolated word recognition is a task that humans can do with much higher accuracy, the isolated word recognition task provides a useful cognitive adversarial attack on deep learning models for speech recognition. It should be acknowledged, however, that the same deep learning models show impressive performance when it comes to the recognition of continuous speech. It seems likely that this excellent performance is due to outstanding top-down language models that have internalized a large proportion of the collective written output of a language. As individual language learners have much less top down knowledge at their individual disposal, they may depend more on the rich information in the speech signal that in the present study we have shown to actually be both present and machine-learnable.

Acknowledgement(s)

The authors thank the Distributed Little Red Hen Lab, co-directed by Francis Steen and Mark Turner, for making the NewsScape corpus available to us, and Peter Uhrig for providing us with the forced alignment results.

Funding

This research was funded by the European Research Council under the ERC grant number 742545, Project WIDE, awarded to the third author.

References

Arnold, D., Tomaschek, F., Lopez, F., Sering, T., and Baayen, R. H. (2017). Words from spontaneous conversational speech can be recognized with human-like accuracy by an error-driven learning algorithm that discriminates between meanings straight from smart acoustic features, bypassing the phoneme as recognition unit. *PLOS ONE*, 12(4):e0174623.

Baayen, R. H. (2005). Data mining at the intersection of psychology and linguistics. In Cutler, A., editor, *Twenty-first century psycholinguistics: Four cornerstones*, pages 69–83. Erlbaum, Hillsdale, New Jersey.

Baayen, R. H., Chuang, Y., and Blevins, J. P. (2018). Inflectional morphology with linear mappings. *The Mental Lexicon*, 13(2):232–270.

Baayen, R. H., Chuang, Y.-Y., Shafaei-Bajestan, E., and Blevins, J. (2019). The discriminative lexicon: A unified computational model for the lexicon and lexical processing in comprehension and production grounded not in (de)composition but in linear discriminative learning. *Complexity*.

Baayen, R. H., Milin, P., Filipović Durdević, D., Hendrix, P., and Marelli, M. (2011). An amorphous model for morphological processing in visual comprehension based on naive discriminative learning. *Psychological Review*, 118:438–482.

Baayen, R. H., Milin, P., and Ramscar, M. (2016a). Frequency in lexical processing. *Aphasiology*, 30(11):1174–1220.

Baayen, R. H., Milin, P., Shaoul, C., Willits, J., and Ramscar, M. (2015). Age of first encounter and age of acquisition norms: What raters do when asked the impossible. *Manuscript, University of Tübingen*.

Baayen, R. H., Shaoul, C., Willits, J., and Ramscar, M. (2016b). Comprehension without segmentation: A proof of concept with naive discriminative learning. *Language, Cognition, and Neuroscience*, 31(1):106–128.

Baayen, R. H. and Smolka, E. (2020). Modelling morphological priming in German with naive discriminative learning. *Frontiers in Communication, section Language Sciences*. preprint on PsyArXiv, doi:10.31234/osf.io/nj39v.

Bitterman, M. (2000). Cognitive evolution: A psychological perspective. In Heyes, C. and Huber, L., editors, *The Evolution of Cognition*, pages 61–79. The MIT Press.

Bruni, E., Tran, N.-K., and Baroni, M. (2014). Multimodal distributional semantics. *Journal of artificial intelligence research*, 49:1–47.

Butterworth, B., editor (1983). *Language Production (Vol. II): Development, Writing and Other Language Processes*. Academic Press, London.

Chomsky, N. and Halle, M. (1968). *The sound pattern of English*. Harper & Row, New York.

Chuang, Y., Vollmer, M.-L., Shafaei-Bajestan, E., Gahl, S., Hendrix, P., and Baayen, R. H. (2020). The processing of nonword form and meaning in production and comprehension: A computational modeling approach using linear discriminative learning. *Behavior Research Methods*.

Chuang, Y.-Y., Loo, K., Blevins, J. P., and Baayen, R. H. (2019). Estonian case inflection made simple. A case study in Word and Paradigm morphology with Linear Discriminative Learning. *PsyArXiv*, pages 1–19.

Cibelli, E. S., Leonard, M. K., Johnson, K., and Chang, E. F. (2015). The influence of lexical statistics on temporal lobe cortical dynamics during spoken word listening. *Brain and language*, 147:66–75.

Connine, C. M., Titone, D., and Wang, J. (1993). Auditory word recognition: Extrinsic and intrinsic effects of word frequency. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 19(1):81.

Dahan, D. and Magnuson, J. S. (2006). Spoken word recognition. In *Handbook of Psycholinguistics*, pages 249–283. Elsevier.

Danks, D. (2003). Equilibria of the Rescorla-Wagner model. *Journal of Mathematical Psychology*, 47(2):109–121.

Diehl, R. L., Lotto, A. J., and Holt, L. L. (2004). Speech perception. *Annu. Rev. Psychol.*, 55:149–179.

Ellis, N. C. (2006). Language acquisition as rational contingency learning. *Applied linguistics*, 27(1):1–24.

Elman, J. L. (2009). On the meaning of words and dinosaur bones: Lexical knowledge without a lexicon. *Cognitive science*, 33(4):547–582.

Ernestus, M. (2000). *Voice assimilation and segment reduction in casual Dutch. A corpus-based study of the phonology-phonetics interface*. LOT, Utrecht.

Fant, G. (1973). *Speech sounds and features*. The MIT Press.

French, N. R., Carter, C. W., and Koenig, W. (1930). The words and sounds of telephone

conversations. *The Bell System Technical Journal*, 9(2):290–324.
Gaskell, M. G. and Marslen-Wilson, W. D. (1997). Integrating form and meaning: A distributed model of speech perception. *Language and cognitive Processes*, 12(5-6):613–656.
Gaskell, M. G. and Marslen-Wilson, W. D. (1999). Ambiguity, competition, and blending in spoken word recognition. *Cognitive Science*, 23(4):439–462.
Gaskell, M. G. and Marslen-Wilson, W. D. (2001). Lexical Ambiguity Resolution and Spoken Word Recognition: Bridging the Gap. *Journal of Memory and Language*, 44(3):325–349.
Gluck, M. A. and Myers, C. E. (2001). *Gateway to memory: An introduction to neural network modeling of the hippocampus and learning*. MIT Press.
Hadamard, J. (1908). *Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées*. Mémoires présentés par divers savants à l'Académie des sciences de l'Institut de France: Éxtrait. Imprimerie nationale.
Harley, T. A. (2013). *The psychology of language: From data to theory*. Psychology press.
Heitmeier, M. and Baayen, R. H. (2020). Simulating phonological and semantic impairment of English tense inflection with Linear Discriminative Learning. *The Mental Lexicon*, accepted. PsyArXiv.
Herdan, G. (1960). *Type-token mathematics: A textbook of mathematical linguistics*, volume 4. Mouton.
Hockett, C. F. and Hockett, C. D. (1960). The origin of speech. *Scientific American*, 203(3):88–97.
Horata, P., Chiewchanwattana, S., and Sunat, K. (2011). A comparative study of pseudo-inverse computing for the extreme learning machine classifier. *The 3rd International Conference on Data Mining and Intelligent Information Technology Applications*, pages 40–45.
Ivens, S. H. and Koslin, B. L. (1991). *Demands for reading literacy require new accountability methods*. Touchstone Applied Science Associates.
Keuleers, E., Stevens, M., Mandera, P., and Brysbaert, M. (2015). Word knowledge in the crowd: Measuring vocabulary size and word prevalence in a massive online experiment. *The Quarterly Journal of Experimental Psychology*, 68(8):1665–1692.
Keune, K., Ernestus, M., Van Hout, R., and Baayen, R. H. (2005). Social, geographical, and register variation in Dutch: From written 'mogelijk' to spoken 'mok'. *Corpus Linguistics and Linguistic Theory*, 1:183–223.
Landauer, T. and Dumais, S. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240.
Landauer, T. K., Foltz, P. W., and Laham, D. (1998). Introduction to latent semantic analysis. *Discourse Processes*, 25:259–284.
Licklider, J. C. (1952). On the process of speech perception. *The journal of the acoustical society of America*, 24(6):590–594.
Long, R. (2018). Enhancing the TASA Corpus for Analysis Using Naive Discriminative Learning. Unpublished MA Thesis Computational Linguistics, University of Tübingen, Tübingen, Germany.
Lu, S.-X., Wang, X., Zhang, G., and Zhou, X. (2015). Effective algorithms of the moore-penrose inverse matrices for extreme learning machine. *Intell. Data Anal.*, 19:743–760.
Luce, P. A., Goldinger, S. D., Auer, E. T., and Vitevitch, M. S. (2000). Phonetic priming, neighborhood activation, and parsyn. *Perception & psychophysics*, 62(3):615–625.
Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.
Magnuson, J. S. (2017). Mapping spoken words to meaning. In Gaskell, M. G. and Mirkovic, J., editors, *Speech Perception and Spoken Word Recognition*, pages 76–96. Routledge, New York.
Magnuson, J. S., You, H., Nam, H., Allopenna, P. D., Brown, K., Escabi, M., Theodore, R. M., Luthra, S., Li, M., and Rueckl, J. (2018). EARSHOT : A minimal neural network model of incremental human speech recognition Frequency. *arXiv*, pages 1–13.
Mandelbrot, B. (1982). *The Fractal Geometry of Nature*. Freeman, San Francisco.

Marslen-Wilson, W. D. (1984). Function and process in spoken word recognition. In *Attention and performance: Control of language processes*, volume X, pages 125–150. Lawrence Erlbaum Associates, Hillsdale, NJ.

Marslen-Wilson, W. D. (1987). Functional parallelism in spoken word-recognition. *Cognition*, 25(1-2):71–102.

Martinet, A. (1967). *Eléments de linguistique générale*. Librairie Armand Colin, 1967.

McClelland, J. L. and Elman, J. L. (1986). The trace model of speech perception. *Cognitive psychology*, 18(1):1–86.

McQueen, J. M. (2005). Speech perception. *The handbook of cognition*, pages 255–275.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Milin, P., Feldman, L. B., Ramscar, M., Hendrix, P., and Baayen, R. H. (2017). Discrimination in lexical decision. *PLOS-one*, 12(2):e0171935.

Milin, P., Madabushi, H. T., Croucher, M., and Divjak, D. (2020). Keeping it simple: Implementation and performance of the proto-principle of adaptation and learning in the language sciences. *arXiv preprint arXiv:2003.03813*.

Morton, J. (1969). Interaction of information in word recognition. *Psychological review*, 76(2):165.

Nixon, J. S. (2020). Of mice and men: Speech sound acquisition as discriminative learning from prediction error, not just statistical tracking. *Cognition*, 197:104081.

Norris, D. (1994a). Shortlist: A connectionist model of continuous speech recognition. *Cognition*, 52(3):189–234.

Norris, D. and McQueen, J. (2008). Shortlist B: A Bayesian model of continuous speech recognition. *Psychological Review*, 115(2):357–395.

Norris, D. G. (1994b). Shortlist: A connectionist model of continuous speech recognition. *Cognition*, 52:189–234.

Olejarczuk, P., Kapatsinski, V., and Baayen, R. H. (2018). Distributional learning is error-driven: the role of surprise in the acquisition of phonetic categories. *Linguistic Vanguard*, 4.

O'shaughnessy, D. (1987). *Speech Communications: Human And Machine (ieee)*. Universities press.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Penrose, R. (1955). A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413.

Phillips, C. (2001). Levels of representation in the electrophysiology of speech perception. *Cognitive Science*, 25(5):711–731.

Pickett, J. and Pollack, I. (1963). Intelligibility of excerpts from fluent speech: Effects of rate of utterance and duration of excerpt. *Language and speech*, 6(3):151–164.

Pisoni, D. B. and Luce, P. A. (1987). Acoustic-phonetic representations in word recognition. *Cognition*, 25(1-2):21—-52.

Plag, I., Homann, J., and Kunter, G. (2017). Homophony and morphology: The acoustics of word-final S in English. *Journal of Linguistics*, 53(1):181–216.

Pollack, I. and Pickett, J. (1963). The intelligibility of excerpts from conversation. *Language and Speech*, 6(3):165–171.

Port, R. F. and Leary, A. P. (2005). Against formal phonology. *Language*, 81:927–964.

Ramscar, M. (2013). Suffixing, prefixing, and the functional order of regularities in meaningful strings. *Psihologija*, 46:377–396.

Ramscar, M., Dye, M., and McCauley, S. M. (2013). Error and expectation in language learning: The curious absence of mouses in adult speech. *Language*, 89(4):760–793.

Ramscar, M., Hendrix, P., Shaoul, C., Milin, P., and Baayen, R. H. (2014). Nonlinear dynamics

of lifelong learning: the myth of cognitive decline. *Topics in Cognitive Science*, 6:5–42.

Ramscar, M. and Yarlett, D. (2007). Linguistic self-correction in the absence of feedback: A new approach to the logical problem of language acquisition. *Cognitive science*, 31(6):927–960.

Ramscar, M., Yarlett, D., Dye, M., Denny, K., and Thorpe, K. (2010). The effects of feature-label-order and their implications for symbolic learning. *Cognitive science*, 34(6):909–957.

Rescorla, R. A. (1988). Pavlovian conditioning: It's not what you think it is. *American Psychologist*, 43(3):151–160.

Rescorla, R. A. and Wagner, A. R. (1972). A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Classical conditioning II: Current research and theory*, 2:64–99.

Scharenborg, O. (2008). Modelling fine-phonetic detail in a computational model of word recognition. In *Proceedings of the Interspeech*. Brisbane, Australia: Causal Productions Pty Ltd.

Scharenborg, O. (2009). Using durational cues in a computational model of spoken-word recognition. *Fundamenta Informaticae - FUIN*, pages 1675–1678.

Scharenborg, O. and Boves, L. (2010). Computational modelling of spoken-word recognition processes: Design choices and evaluation. *Pragmatics & Cognition*, 18(1):136–164.

Seidenberg, M. S. and McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological review*, 96(4):523.

Sering, T., Milin, P., and Baayen, R. H. (2018). Language comprehension as a multiple label classification problem. *Statistica Neerlandica*, pages 1–15.

Shafaei-Bajestan, E. and Baayen, H. (2020). Wide learning of the comprehension of morphologically complex words: from audio signal to semantics.

Shafaei-Bajestan, E. and Baayen, R. H. (2018). Wide learning for auditory comprehension. In Yegnanarayana, B., editor, *Proceedings of Interspeech 2018*, pages 966–970, Hyderabad, India: International Speech Communication Association (ISCA).

Siegel, S. and Allan, L. G. (1996). The widespread influence of the rescorla-wagner model. *Psychonomic Bulletin & Review*, 3(3):314–321.

ten Bosch, L., Boves, L., Tucker, B., and Ernestus, M. (2015). Diana: towards computational modeling reaction times in lexical decision in north american english. In *Interspeech 2015: 16th Annual Conference of the International Speech Communication Association*, pages 1576–1580. Dresden: International Speech Communication Association.

Tomaschek, F., Plag, I., Ernestus, M., and Baayen, R. H. (2019). Modeling the duration of word-final s in english with naive discriminative learning. *Journal of Linguistics*. https://psyarxiv.com/4bmwg, doi = 10.31234/osf.io/4bmwg.

Tucker, B. V., Brenner, D., Danielson, K., Kelley, M. C., Nenadić, F., and Sims, M. (2017). The massive auditory lexical decision database: Toward reliable, generalizable speech research. *Manuscript submitted for publication*.

Warren, R. M. (1970). Perceptual restoration of missing speech sounds. *Science*, 167(3917):392–393.

Warren, R. M. (1971). Identification times for phonemic components of graded complexity and for spelling of speech. *Perception & Psychophysics*, 9(4):345–349.

Warren, R. M. (2000). Phonemic organization does not occur: Hence no feedback. *Behavioral and Brain Sciences*, 23(3):350–325.

Weber, A. and Scharenborg, O. (2012). Models of spoken-word recognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, 3(3):387–401.

Weitz, M. (2019). Balancing bias in natural language recognition using LSTMs. Unpublished Lab Rotation Report at Quantitative Linguistics Group, University of Tübingen, Tübingen, Germany.

Widrow, B. and Hoff, M. E. (1960). Adaptive switching circuits. *1960 WESCON Convention Record Part IV*, pages 96–104.

Wieling, M., Nerbonne, J., Bloem, J., Gooskens, C., Heeringa, W., and Baayen, R. H. (2014). A cognitively grounded measure of pronunciation distance. *PLOS-ONE*, 9(1):e75734.

Wood, S. N. (2017). *Generalized Additive Models.* Chapman & Hall/CRC, New York.

Appendix A. Lemmas

Lemma A.1. *Let the function E_t be defined according to equation 1. Then*

$$\frac{\partial E_t}{\partial w_{tij}} = -(y_{tj} - \hat{y_{tj}}) f'(a_{tj}) x_{ti}.$$

Proof

$$\begin{aligned}
\frac{\partial E_t}{\partial w_{tij}} &= \frac{\partial\big(\sum_j \frac{1}{2}(y_{tj} - \hat{y_{tj}})^2\big)}{\partial w_{tij}} && \text{definition of } E_t \\
&= \frac{\partial\big(\frac{1}{2}(y_{tj} - \hat{y_{tj}})^2\big)}{\partial w_{tij}} && \text{error for neuron } j \\
&= \frac{\partial\big(\frac{1}{2}(y_{tj} - \hat{y_{tj}})^2\big)}{\partial \hat{y_{tj}}} \frac{\partial \hat{y_{tj}}}{\partial w_{tij}} && \text{chain rule} \\
&= -(y_{tj} - \hat{y_{tj}}) \frac{\partial \hat{y_{tj}}}{\partial w_{tij}} && \text{partial derivative} \\
&= -(y_{tj} - \hat{y_{tj}}) \frac{\partial \hat{y_{tj}}}{\partial a_{tj}} \frac{\partial a_{tj}}{\partial w_{tij}} && \text{chain rule} \\
&= -(y_{tj} - \hat{y_{tj}}) \frac{\partial f(a_{tj})}{\partial a_{tj}} \frac{\partial a_{tj}}{\partial w_{tij}} && \text{definition of } \hat{y_{tj}} \\
&= -(y_{tj} - \hat{y_{tj}}) f'(a_{tj}) \frac{\partial a_{tj}}{\partial w_{tij}} \\
&= -(y_{tj} - \hat{y_{tj}}) f'(a_{tj}) \frac{\partial(\sum_k x_{tk} w_{tkj})}{\partial w_{tij}} && \text{definition of } a_{tj} \\
&= -(y_{tj} - \hat{y_{tj}}) f'(a_{tj}) \sum_k \frac{\partial(x_{tk} w_{tkj})}{\partial w_{tij}} && \text{properties of summation} \\
&= -(y_{tj} - \hat{y_{tj}}) f'(a_{tj}) \frac{\partial(x_{ti} w_{tij})}{\partial w_{tij}} && \text{zero for all } k \neq i \\
&= -(y_{tj} - \hat{y_{tj}}) f'(a_{tj}) x_{ti}. && \text{partial derivative}
\end{aligned}$$

Lemma A.2. *Using the variables defined in section 3.2, let $\boldsymbol{x}_t \in \{0,1\}^m$, $\boldsymbol{y}_t \in \{0,1\}^n$. Let f be the identity function for all o_j. Let $\eta = \alpha\beta$. Then, the learning rule of Rescorla-Wagner can be written as the following*

$$
\begin{aligned}
\Delta w_{tij} &= \eta(y_{tj} - \hat{y_{tj}}) f'(a_{ti}) x_{ti} && \textit{Equation 2} \\
&= \eta(y_{tj} - \hat{y_{tj}}) \frac{\partial a_{ti}}{\partial a_{ti}} x_{ti} && \textit{identity function} \\
&= \eta(y_{tj} - \sum_i x_{ti} w_{tij}) x_{ti}. && \textit{defintion of } \hat{y_{tj}} \\
&= \begin{cases} \eta(y_{tj} - \sum_i x_{ti} w_{tij}), & \textit{if } x_{ti} = 1; \\ 0, & \textit{if } x_{ti} = 0. \end{cases} && x_{ti} \textit{ is binary} \\
&= \begin{cases} \eta(1 - \sum_i x_{ti} w_{tij}), & \textit{if } x_{ti} = 1 \textit{ and } y_{tj} = 1; \\ \eta(0 - \sum_i x_{ti} w_{tij}), & \textit{if } x_{ti} = 1 \textit{ and } y_{tj} = 0; \\ 0, & \textit{if } x_{ti} = 0. \end{cases} && y_{tj} \textit{ is binary}
\end{aligned}
$$

Lemma A.3. *Let $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{W}$ be three matrices such that $\boldsymbol{XW} = \boldsymbol{Y}$, and let $\boldsymbol{X}$ be an m-by-m invertible matrix. Then $\boldsymbol{W} = \boldsymbol{X}^{-1}\boldsymbol{Y}$*

$$
\begin{aligned}
\boldsymbol{XW} &= \boldsymbol{Y} && \\
\boldsymbol{X}^{-1}\boldsymbol{XW} &= \boldsymbol{X}^{-1}\boldsymbol{Y} && \textit{left multiply both sides with } \boldsymbol{X}^{-1} \\
\boldsymbol{I}_m\boldsymbol{W} &= \boldsymbol{X}^{-1}\boldsymbol{Y} && \boldsymbol{X}^{-1}\boldsymbol{X} = \boldsymbol{I}_m \\
\boldsymbol{W} &= \boldsymbol{X}^{-1}\boldsymbol{Y} && \boldsymbol{I}_m\boldsymbol{W} = \boldsymbol{W}
\end{aligned}
$$