Discriminative learning and the lexicon: NDL and LDL

Yu-Ying Chuang and R. Harald Baayen

University of Tübingen, Germany

Table of Contents

Summary

NDL and LDL are simple computational algorithms for lexical learning and lexical processing. Both NDL and LDL assume that learning is discriminative, driven by prediction error, and that it is this error which calibrates the association strength between input and output representations. Both words' forms and their meanings are represented by numeric vectors, and mappings between forms and meanings are set up. For comprehension, form vectors predict meaning vectors. For production, meaning vectors map onto form vectors. These mappings can be learned incrementally, approximating how children learn the words of their language. Alternatively, optimal mappings representing the endstate of learning can be estimated. The NDL and LDL algorithms are incorporated in a computational theory of the mental lexicon, the 'discriminative lexicon'. The model shows good performance both with respect to production and comprehension accuracy, and for predicting aspects of lexical processing, including morphological processing, across a wide range of experiments. Since mathematically, NDL and LDL implement multivariate multiple regression, the 'discriminative lexicon' provides a cognitively motivated statistical modeling approach to lexical processing.

Keywords

incremental learning, computational modeling, word and paradigm morphology, mental lexicon, lexical processing, language comprehension, speech production, linear discriminative learning, naive discriminative learning

1. Introduction

A wide range of algorithms has been explored and tested to better understand language learning (see, e.g., Kapatsinski, 2018, for an overview). One particular learning algorithm has proved particularly useful for understanding both first language acquisition (Ramscar et al., 2010, 2013a) and second language acquisition (Ellis, 2006, 2013). This learning algorithm builds on a learning rule introduced by Rescorla and Wagner (1972). It formalizes how we learn from the errors we make. Models that build on this learning rule are a special instance of a more general class of supervised, error-driven learning algorithms in machine learning. In what follows, following Ramscar et al. (2010), we refer to models building on the learning rule of Rescorla and Wagner as discriminative learning models. The kind of error-driven learning that we are particularly interested in is subliminal and involves an on-going fine-tuning of the learning system that continues across the lifespan (Ramscar et al., 2014), without requiring conscious awareness. The difference between this low-level implicit learning and high-level conscious reasoning in learning is brought out clearly by Ramscar et al. (2013a), who showed that when presented with the same learning materials, decisions made by young children differ markedly from those made by adults.

Regardless of whether they have to learn tones in Mandarin or Vietnamese, or case inflections in Finnish or Estonian, young children have little difficulty picking up the regularities in their language without being bothered by the many irregularities that are also present. As children grow up, through education, more and more high-level linguistic knowledge is introduced to them, which can be very beneficial especially for second language learning. What discriminative learning seeks to approximate, however, is how children learn their native languages before being explicitly taught about the grammar of these languages. Importantly, this kind of low-level learning does not stop at puberty or at reaching adulthood. It is an ongoing process that fine-tunes our cognitive system to our environment, without demanding or requiring conscious attention.

In what follows, we address the discriminative learning of word knowledge, focusing on the algorithms of two related computational models for lexical processing in the mental lexicon, Naïve

Discriminative Learning (NDL) and Linear Discriminative Learning (LDL), both of which take inspiration from two related learning rules, the learning rule of Rescorla and Wagner (1972) and the learning rule of Widrow and Hoff (1960). In Section 2, we first introduce the Rescorla-Wagner learning rule and the algorithms that build on it, and explain their relation to regression in statistics and the concept of proportional analogy in Word and Paradigm Morphology (Matthews, 1974; Blevins, 2016). Subsequently in Section 3, we present a computational blueprint of the mental lexicon that covers both comprehension and production. Section 4 provides an overview of the results obtained with NDL/LDL. The section on internal validation addresses how accurate the model is when given the task to understand or produce words. The section on external validation provides an overview of how the model has been used to predict aspects of lexical processing. Finally, in Section 5, we discuss the pros and cons of NDL and LDL compared to statistical modeling and machine learning.

2. Discriminative learning

Consider how English-speaking children learn the word *cat*. When English-speaking children hear the sequence of sounds /k/, /æ/ and /t/, they are likely to think of a four-legged furry animal with whiskers and a tail. In what follows, we first represent the meaning of cat symbolically, by a unique semantic unit. This is an obvious simplification which implies that all meanings are completely unrelated. Later, we will relax this simplifying assumption, thereby clearing the way for observing and modeling interactions between form and meaning (see, e.g., Heitmeier and Baayen, 2021). In our modeling framework, we refer to the basic elements of meaning, for which a semantic representation is posited, as “lexomes”, which will henceforth be represented by capital letters (e.g., CAT). Over time, children gradually learn to associate the sound sequence /kæt/ with the lexome CAT, using form to discriminate between possible meanings. When the association strength of /kæt/ and CAT asymptotes, the word has been fully “learned”. As shown in the left part of Figure 1, the connection weights between /k/, /æ/, /t/ and CAT steadily increase together with learning and then reach asymptote.

Association learning, or Hebbian learning, however, is only part of discriminative learning. What critically distinguishes discriminative learning from Hebbian learning (see Rescorla, 1988) is that whenever the associations between the sound sequence /k/, /æ/, /t/ and CAT are strengthened, the associations between /k/, /æ/, /t/ and other meanings (e.g., KEY, DOG, BALL, TREE, etc.) are weakened at the same time. In other words, every "correct" learning of /k/, /æ/, /t/ with CAT simultaneously implies "unlearning" associations between /k/, /æ/, /t/ with all the other word meanings. By way of example, after 200 learning events of *cat*, we can let learning continue with the addition of a second word, *key*. While the weights between /æ/, /t/ and CAT (blue and red solid lines) remain high throughout subsequent learning, the weights between /æ/, /t/ and KEY (blue and red dotted lines), on the other hand, which start at zero, become increasingly negative. Turning to the /i/, which is unique to the word *key*, we see that its weight to KEY (grey dotted line) increases rapidly, whereas the weight on the connection between /i/ to CAT (grey solid line) is progressively weakened.

What is also interesting is the development of the connection weights for /k/, which is shared by both words. For CAT, unlike what we find for /æ/ and /t/, the weight between /k/ and CAT (orange solid line) actually starts to decrease once the word *key* is encountered. This is not what is predicted by Hebbian learning, since /k/ and CAT, which always occur together, should always be strengthened. As for KEY, the weight between /k/ and KEY (orange dotted line) does not grow as strong as that between /i/ and KEY. This illustrates a crucial aspect of discriminative learning: cue competition. Since /k/ is a cue for both CAT and KEY, it is therefore not a very reliable cue for distinguishing between the meanings of these two words. On the other hand, /æ/, /t/, and /i/ are cues with high discriminability: they unambiguously point to CAT and KEY respectively.

The simplified example presented in Figure 1 illustrates how the core learning engine works that the theory of the discriminative lexicon makes use of to model mappings between forms and meanings. This learning engine can be represented by a fully-connected two-layer network, with every cue (phones in the present example) connected with every outcome (meanings), as shown in Figure 2. Importantly, the network is dynamic, as weights in the network are constantly updated according to experience, as illustrated for the development of weights in Figure 1. Learning, in this

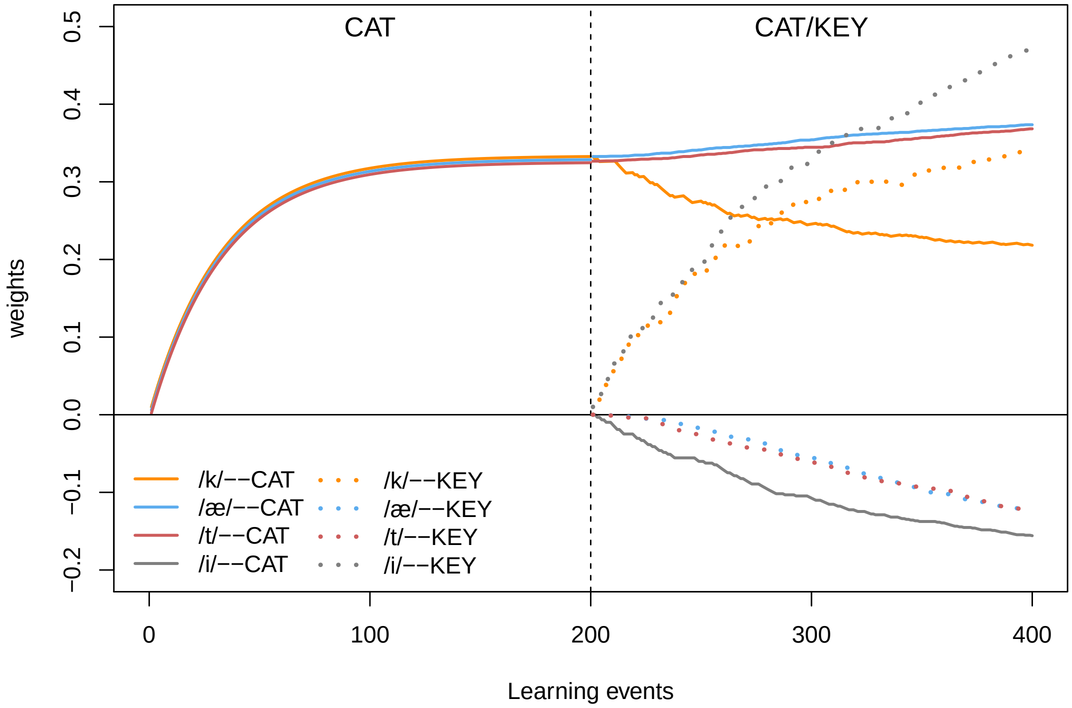

Figure 1: The development of connection weights between word form cues /k/, /æ/, /t/, and /i/ and word meanings CAT and KEY. The first half of the learning only involves CAT, whereas the second half involves CAT and KEY presented at equal rate. As learning progresses, phone cues develop increasingly positive connection weights with their target meaning outcomes (e.g., /i/ to KEY), and simultaneously increasingly negative weights with the non-targeted meaning (e.g., /i/ to CAT). For phone cues that are not unique to a given outcome (i.e., /k/), the connection weights are calibrated more substantially due to cue competition.

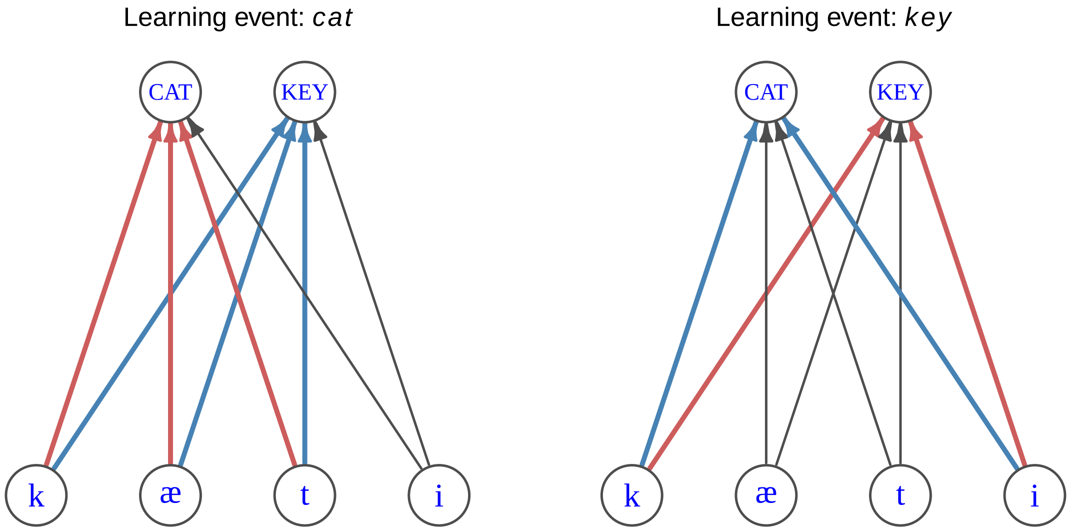

Figure 2: The fully-connected network between word form cues (/k/, /æ/, /t/, and /i/) and meaning outcomes (CAT, KEY) that underlies Figure 1. When the learning event involves *cat* (left-hand panel), for cues /k/, /æ/, /t/, their connection weights to CAT are strengthened (indicated by red arrows), while those to KEY are weakened (indicated by blue arrows). Since /i/ is not involved in the learning event of *cat*, its weight to either outcome remains unchanged (indicated by black arrows). On the other hand, when *key* is learned (right-hand panel), the weights of /k, i/ to KEY are strengthened, and those to CAT are weakened. The efferent weights from /æ, t/ remain unchanged.

approach, is understood as a continuous process of weight recalibration that continues as long as a language is used. This subliminal and implicit learning is taking place all the time, not only for children, but also for adults. For example, Ramscar et al. (2014) investigated aging adults' lexical processing and found that what has been commonly considered as cognitive decline is actually a consequence of continuous language learning. In Section 4.3.3 we will see even within a large lexical decision experiment, learning is going on and can be traced with discriminative learning.

In the following sections, we first present the formalizations of the learning algorithm, covering both how to model incremental learning using the Rescorla-Wagner rule (Section 2.1), and how to estimate the endstate of learning using matrix algebra (Section 2.2). In Section 2.3 we illustrate how to model morphology in this framework, and discuss the parallelism between the model and proportional analogy in Word and Paradgim Morphology.

2.1 Formalizing the learning algorithm

The algorithms underlying discriminative learning are given by the Rescorla-Wagner learning rule (Rescorla and Wagner, 1972) and the Widrow-Hoff learning rule (Widrow and Hoff, 1960). In fact, the two learning rules are closely related, and have been widely applied in both psychology and physics. In the domain of language learning, Ramscar and Yarlett (2007), Ramscar et al. (2010) and Ellis (2006, 2013), for example, use the Rescorla-Wagner rule to explain phenomena observed in first and second language acquisition. The core of the rules for weight updating can be straightforwardly formalized as:

$$w_{ij}^{t+1} = w_{ij}^{t} + \Delta w_{ij}^{t}. \tag{1}$$

Equation (1) specifies that for a given weight on the connection between cue$_i$ and outcome$_j$ (e.g., between /k/ and CAT), the new weight after updating (at time $t+1$) is the sum of the current weight (at time t) and an adjustment Δw_{ij}^{t}. The formal definition of the adjustment Δw_{ij}^{t} in a formulation of the Rescorla-Wagner learning rule that is most similar to the Widrow-Hoff learning rule is as follows:

$$\Delta w_{ij}^{t} \quad = \quad \begin{cases} 0 & \text{if } \textsc{absent}(C_i, t) \\ \alpha \left(\lambda - \sum_{\textsc{present}(C_k,\, t)} w_{kj}\right) & \text{if } \textsc{present}(C_i, t) \;\&\; \textsc{present}(O_j, t) \\ \alpha \left(0 - \sum_{\textsc{present}(C_k,\, t)} w_{kj}\right) & \text{if } \textsc{present}(C_i, t) \;\&\; \textsc{absent}(O_j, t) \end{cases} \tag{2}$$

Learning rule (2) specifies that when a given cue (C_i) is absent in the current learning event, Δw_{ij}^{t} is 0: the weights on the efferent connections of this cue are left untouched. Thus, when the learning event comprises the cues /k/, /æ/, and /t/ and the outcome CAT, the weights from /i/ to all the outcomes will not be changed. On the other hand, when a cue C_i is present in the current learning event, Δw_{ij}^{t} equals the prediction error times the learning rate α. (In our simulations, α is set to a low number, either 0.01 or 0.001.) When cue C_i is present and outcome O_j is also present, then the prediction

Table 1: Weight adjustment for all the connections between phone cues /k/, /æ/, /t/, and /i/ and meaning outcomes CAT and KEY after a learning event of *cat*.

(a) Weights w_{ij}^{t} at t

	CAT	KEY
/k/	0.1	0.3
/æ/	0.3	-0.1
/t/	0.3	-0.1
/i/	-0.2	0.4

(b) Weight adjustment Δw_{ij}^{t}

	CAT	KEY
/k/	0.03	-0.01
/æ/	0.03	-0.01
/t/	0.03	-0.01
/i/	0.00	0.00

(c) Weights w_{ij}^{t+1} at $t+1$

	CAT	KEY
/k/	0.13	0.29
/æ/	0.33	-0.11
/t/	0.33	-0.11
/i/	-0.20	0.40

error is defined by λ (which is always set to 1 in our simulations) minus the predicted support for outcome O_j. This predicted support is defined as the sum of the weights on the connections from each of the cues (indexed by k) that are in the input at time t to outcome O_j. When outcome O_j is absent, the prediction error is given by 0 minus this predicted support. In this way, the model learns from its mistakes.

To make this more concrete, consider Table 1a, which presents the respective connection weights of /k/, /æ/, /t/, and /i/ to CAT and KEY at a given time point t. For a learning event of CAT (with cues /k/, /æ/, and /t/), the predicted support of these cues for CAT is $1 \times 0.1 + 1 \times 0.3 + 1 \times 0.3 = 0.7$. With the learning rate set to 0.1, for each of these cues, the adjustment of the weights on their connection to CAT, Δw, is equal to $0.1 \times (1 - 0.7) = 0.03$. In the same vein, the predicted support of these cues for KEY is $1 \times 0.3 + 1 \times -0.1 + 1 \times -0.1 = 0.1$. Since KEY is not the present outcome in this learning event, the adjustment to the weights on the connections from these cues to KEY is Δw = $0.1 \times (0 - 0.1) = -0.01$. Thus, after this learning event, the weights between /k/, /æ/, and /t/ to CAT will all increase by 0.03, whereas the weights between /k/, /æ/, and /t/ to KEY will all decrease by 0.01. The weights on the efferent connections of /i/ to CAT and KEY, on the other hand, remain unchanged because the cue /i/ is not involved in this learning event (Table 1b and 1c, see also Figure 2). The weight developments presented in Figure 1 are obtained by applying this learning algorithm incrementally, starting with initial weights that are all zero.

Discriminative learning is used by two related models. Naïve Discriminative Learning (NDL, Baayen et al., 2011) implements the Rescorla-Wagner rule, using binary coding to represent the presence and absence of cues and outcomes. Linear Discriminative Learning (LDL, Baayen et al.,

2019) allows cues and outcomes to be represented by vectors of real numbers. LDL therefore has to make use of the Widrow-Hoff learning rule. For a mathematical discussion of the Widrow-Hoff learning rule and its relation to the Rescorla-Wagner rule, see Milin et al. (2020) and Shafaei-Bajestan et al. (2020).

2.2 Estimating the endstate of learning

When learning continues indefinitely, weights between cues and outcomes reach an equilibrium state, where the amount by which weights change is so small that they become negligible. We refer to this state as the theoretical endstate of learning (henceforth EoL). In NDL, the EoL is estimated with the Danks equations (Danks, 2003), while in LDL, the EoL is estimated with the mathematics of multivariate multiple regression. Thus, instead of iterating the learning algorithm over thousands or millions of learning events, we can estimate EoL weights directly. The Danks equations compute the estimated weights from the conditional probabilities of cue and outcome combinations. In what follows, we explain how LDL estimates the EoL.

Continuing with our toy example, we first create a word form matrix $\boldsymbol{C}$ and a semantic matrix $\boldsymbol{S}$, as shown in (3) and (4). For readers unfamiliar with vectors and matrices, a brief introduction is provided in the note entitled ‘Vectors and matrix multiplication’ at the end of this article.

$$\boldsymbol{C} = \begin{array}{c} \\ \textit{cat} \\ \textit{key} \end{array} \!\!\begin{array}{c} \begin{array}{cccc} \text{k} & \text{æ} & \text{t} & \text{i} \end{array} \\ \left(\begin{array}{cccc} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{array} \right) \end{array}, \tag{3}$$

$$\boldsymbol{S} = \begin{array}{c} \\ \textit{cat} \\ \textit{key} \end{array} \!\!\begin{array}{c} \begin{array}{cc} \text{CAT} & \text{KEY} \end{array} \\ \left(\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right) \end{array}. \tag{4}$$

For both matrices, rows represent individual words. In this example we use binary encoding, in which word form cues or meanings that are present in a given word are coded as ones, and those that are absent are coded as zeros. However, the row vectors of $\boldsymbol{C}$ and $\boldsymbol{S}$ can also be real-valued vectors. Given $\boldsymbol{C}$ and $\boldsymbol{S}$, we can predict word meanings from word forms (comprehension), and word forms from word meanings (production). The mappings of comprehension and production can be formalized with Equations (5) and (6). The matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ are the weight matrices of the comprehension and production networks respectively.

$$\begin{aligned} \boldsymbol{CF} &= \boldsymbol{S}, && (5) \\ \boldsymbol{SG} &= \boldsymbol{C}. && (6) \end{aligned}$$

$\boldsymbol{F}$ and $\boldsymbol{G}$ can be obtained by solving Equations 7 and 8.

$$\begin{aligned} \boldsymbol{F} &= (\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{S}, && (7) \\ \boldsymbol{G} &= (\boldsymbol{S}^T\boldsymbol{S})^{-1}\boldsymbol{S}^T\boldsymbol{C}. && (8) \end{aligned}$$

Since the matrices $(\boldsymbol{C}^T\boldsymbol{C})$ and $(\boldsymbol{S}^T\boldsymbol{S})$ can be singular, we use the Moore-Penrose pseudo-inverse to calculate their inverse matrices. Returning to our toy example, $\boldsymbol{F}$, the estimated weight matrix of the comprehension network, is:

$$\boldsymbol{F} = \begin{array}{c} \\ \text{k} \\ \text{æ} \\ \text{t} \\ \text{i} \end{array} \begin{array}{c} \begin{array}{cc} \text{CAT} & \text{KEY} \end{array} \\ \begin{pmatrix} 0.2 & 0.4 \\ 0.4 & -0.2 \\ 0.4 & -0.2 \\ -0.2 & 0.6 \end{pmatrix} \end{array}. \qquad (9)$$

The numbers in $\boldsymbol{F}$ are exactly the same as those in the weight matrix obtained by applying the

Rescorla-Wagner learning rule (2) incrementally over 10,000 learning events, with the two outcomes appearing randomly and roughly equally often over intervals of time.

Interestingly, the mathematics underlying LDL is identical to that underlying multivariate multiple regression. Suppose that we have a design matrix $\boldsymbol{X}$ with n observations and p predictors, as shown in (10) (the first column represents the intercept).

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix} \tag{10}$$

For example, for a set of words represented on the rows of the matrix, we have values for predictors such as their length, stress position, and frequency of occurrences. In standard multiple regression, we have only one response variable, e.g., reaction times in a lexical decision task. In multivariate multiple regression, however, there are multiple response variables. For instance, in addition to reaction times, we may also have other response variables such as error rates, acoustic durations in a production task, etc. For k response variables, the response matrix $\boldsymbol{Y}$ has the following form:

$$\boldsymbol{Y} = \begin{pmatrix} y_{11} & y_{12} & \dots & y_{1k} \\ y_{21} & y_{22} & \dots & y_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ y_{n1} & y_{n2} & \dots & y_{nk} \end{pmatrix} \tag{11}$$

In order to predict $\boldsymbol{Y}$ from $\boldsymbol{X}$, multivariate multiple regression estimates a coefficient matrix $\boldsymbol{B}$,

$$\boldsymbol{B} = \begin{pmatrix} \beta_{01} & \beta_{02} & \dots & \beta_{0k} \\ \beta_{11} & \beta_{12} & \dots & \beta_{1k} \\ \vdots & \vdots & \vdots & \vdots \\ \beta_{p1} & \beta_{p2} & \dots & \beta_{pk} \end{pmatrix}, \tag{12}$$

by solving

$$\boldsymbol{XB} = \boldsymbol{Y}, \tag{13}$$

which is of exactly the same form as equations (5) and (6). Note that a predicted response value $\hat{y}_{ij}$ is obtained by pairwise multiplication of the i-th row of $\boldsymbol{X}$ and the j-th column of $\boldsymbol{B}$:

$$\hat{y}_{ij} = \beta_{0j} + x_{i1}\beta_{1j} + x_{i2}\beta_{2j} + \ldots + x_{ip}\beta_{pj} \tag{14}$$

In summary, to estimate the EoL, LDL makes use of the same mathematics as multivariate multiple regression, with the only difference being that an intercept term is not included. Thus, the column vector for /k/ in the form matrix $\boldsymbol{C}$ (3) can be seen as giving the values of a predictor variable, and the CAT column in the semantic matrix $\boldsymbol{S}$ (4) can be understood as specifying the values of a response variable. It is important to note that, just as in simple linear regression, a regression line cannot go through all datapoints but is estimated such that the error is minimized, the predictions of LDL are approximate, but optimal in the least squares sense.

2.3 Morphology

In many languages, words are inflected for categories such as tense, aspect, mood, person, case, and number. English *talked* describes a communication event that happened in the past. In languages with more elaborate morphological systems, more than one inflectional meaning can be realized in an inflected form. In Estonian, for example, *jalgadel* is the plural form of the adessive case of the noun *jalg* 'foot', meaning 'on the feet'. The way that NDL and LDL deal with inflected words is to construct the meaning of an inflected word from the meaning of its base word and the pertinent inflectional meanings.

NDL uses one-hot encoding to represent both content and inflectional meanings. As shown in the upper panel of Table 2, for *talked*, the meanings of TALK and PAST are both coded with a 1, and the vector for *talked* is simply the sum of the vectors of stem and inflection. In this way, an inflected word comes to be represented by a binary vector in a high-dimensional semantic space, the dimension of which is given by the number of different word meanings and inflectional meanings.

This one-hot encoding mechanism for lexomes captures the semantic similarity between different inflected forms of the same base word, and between different words with the same inflection, but it fails to capture the semantic similarity between different base words. *Dog* and *cat* are equally unrelated semantically as *dog* and *key*. To address this issue, LDL, inspired by distributional semantics (Landauer and Dumais, 1997), represents meanings with vectors of real numbers. Ideally, these vectors are derived from corpora, in which case *dog* and *cat* will have more similar vectors than *dog* and *key*. Exactly as in NDL, LDL defines the vector for a complex word to be the sum of the vectors of its content and inflectional lexomes, as illustrated in the bottom part of Table 2.

Table 2: Semantic vector representations for inflected words in NDL (top) and LDL (bottom).

					NDL					
	L1	L2	L3	PAST	L5	TALK	L7	L8	L9	L10
talk	0	0	0	0	0	1	0	0	0	0
past	0	0	0	1	0	0	0	0	0	0
talked	0	0	0	1	0	1	0	0	0	0
					LDL					
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
talk	-0.16	0.25	-0.07	-0.04	0.13	0.03	-0.04	0.10	0.22	-0.25
past	0.16	0.13	-0.36	0.01	-0.07	-0.04	-0.32	0.07	0.45	0.14
talked	0.00	0.38	-0.43	-0.03	0.06	-0.01	-0.36	0.17	0.67	-0.11

Importantly, this way of representing words' meanings, combined with the algorithm of discriminative learning, formalizes the concept of proportional analogy in Word and Paradigm Morphology (WPM, Matthews, 1974; Blevins, 2016). In contrast to traditional morphological theories that regard morphemes as the smallest meaningful units of the grammar, WPM takes words to be simplest cognitively valid representational units. Accordingly, inflected forms are not put together by concatenating morphemes, but are obtained by means of proportional analogy. According to proportional analogy, given the relation between *talk* and *talked*, English speakers will be able to infer the past tense form for *kick* (*talk* : *talked* = *kick* : ***kicked***). Within the present framework, we can formalize this analogy as follows. We begin with treating *talk*, *talked*, and *kick* as given information (first three rows in Table 3), and train a production network on these words. With the

Table 3: A toy example of forming English past tense verbs by using proportional analogy.

Base meaning	Inflectional meaning	Form
TALK	PRESENT	tɔk
TALK	PAST	tɔkt
KICK	PRESENT	kɪk
KICK	PAST	kɪkt

trained network, we can then test whether *kicked* is correctly produced (last row in Table 3).

We first build the semantic representations for the three known words by summing the semantic vectors of the base words and the pertinent semantic vectors for tense.

$$\begin{aligned}\overrightarrow{\text{TALK.PRES}} &= \overrightarrow{\text{TALK}} + \overrightarrow{\text{PRESENT}},\\ \overrightarrow{\text{TALK.PAST}} &= \overrightarrow{\text{TALK}} + \overrightarrow{\text{PAST}},\\ \overrightarrow{\text{KICK.PRES}} &= \overrightarrow{\text{KICK}} + \overrightarrow{\text{PRESENT}}.\end{aligned} \tag{15}$$

This gives us the semantic matrix $\boldsymbol{S}$:

$$\boldsymbol{S} = \begin{array}{c} \\ talk \\ talked \\ kick \end{array} \begin{array}{c} \begin{array}{rrrrrrrrr} \text{S1} & \text{S2} & \text{S3} & \text{S4} & \text{S5} & \text{S6} & \text{S7} & \text{S8} & \text{S9} \end{array} \\ \left(\begin{array}{rrrrrrrrr} 10.58 & 6.97 & 8.94 & -3.60 & 1.16 & -11.02 & -4.92 & -1.93 & 1.09 \\ -0.63 & -4.04 & 13.60 & -4.09 & 1.53 & 2.07 & -1.17 & 3.47 & 2.41 \\ 1.98 & 3.04 & -0.56 & -1.19 & -0.16 & -8.06 & -12.07 & -4.58 & -11.59 \end{array}\right) \end{array}. \tag{16}$$

For the word form matrix $\boldsymbol{C}$, we use biphones instead of uniphones. The word *talk*, for example, contains the biphones #t, tɔ, ɔk, and k#, with # indicating word boundaries.

$$\boldsymbol{C} = \begin{array}{c} \\ \textit{talk} \\ \textit{talked} \\ \textit{kick} \end{array} \begin{array}{c} \begin{array}{ccccccccc} \text{\#t} & \text{tɔ} & \text{ɔk} & \text{k\#} & \text{kt} & \text{t\#} & \text{\#k} & \text{kɪ} & \text{ɪk} \end{array} \\ \left(\begin{array}{ccccccccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{array}\right) \end{array}. \tag{17}$$

Given $\boldsymbol{S}$ and $\boldsymbol{C}$, using Equation 8 we obtain the production network $\boldsymbol{G}$:

$$\boldsymbol{G} = \begin{array}{c} \\ \text{S1} \\ \text{S2} \\ \text{S3} \\ \text{S4} \\ \text{S5} \\ \text{S6} \\ \text{S7} \\ \text{S8} \\ \text{S9} \end{array} \begin{array}{c} \begin{array}{ccccccccc} \text{\#t} & \text{tɔ} & \text{ɔk} & \text{k\#} & \text{kt} & \text{t\#} & \text{\#k} & \text{kɪ} & \text{ɪk} \end{array} \\ \left(\begin{array}{ccccccccc} 0.02 & 0.02 & 0.02 & 0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 \\ 0.00 & 0.00 & 0.00 & 0.02 & -0.03 & -0.03 & -0.01 & -0.01 & -0.01 \\ 0.06 & 0.06 & 0.06 & 0.01 & 0.05 & 0.05 & 0.00 & 0.00 & 0.00 \\ -0.02 & -0.02 & -0.02 & -0.01 & -0.02 & -0.02 & 0.00 & 0.00 & 0.00 \\ 0.01 & 0.01 & 0.01 & 0.00 & 0.01 & 0.01 & 0.00 & 0.00 & 0.00 \\ -0.01 & -0.01 & -0.01 & -0.03 & 0.02 & 0.02 & 0.00 & 0.00 & 0.00 \\ -0.01 & -0.01 & -0.01 & -0.03 & -0.02 & -0.02 & -0.04 & -0.04 & -0.04 \\ 0.01 & 0.01 & 0.01 & -0.01 & 0.01 & 0.01 & -0.01 & -0.01 & -0.01 \\ 0.02 & 0.02 & 0.02 & -0.02 & -0.01 & -0.01 & 0.04 & -0.04 & -0.04 \end{array}\right) \end{array}. \tag{18}$$

This weight matrix specifies the weights from the semantic dimensions (S1, S2, …, S9) to the biphones.

To produce the past tense form of *kick*, we first define the corresponding semantic vector ($\boldsymbol{s}_{\text{kicked}}$) by summing the vectors of KICK and PAST ($\overrightarrow{\text{KICK.PAST}} = \overrightarrow{\text{KICK}} + \overrightarrow{\text{PAST}}$), resulting in

$$\boldsymbol{s}_{\text{kicked}} = \begin{array}{c} \begin{array}{ccccccccc} \text{S1} & \text{S2} & \text{S3} & \text{S4} & \text{S5} & \text{S6} & \text{S7} & \text{S8} & \text{S9} \end{array} \\ \left(\begin{array}{ccccccccc} -7.68 & -7.13 & 2.80 & -2.56 & 0.31 & 1.91 & -8.29 & -0.63 & -10.44 \end{array}\right) \end{array}. \tag{19}$$

With the network $\boldsymbol{G}$, we can now predict the form vector for *kicked*:

$$\hat{\boldsymbol{c}}_{\text{kicked}} = \boldsymbol{s}_{\text{kicked}}\boldsymbol{G}, \tag{20}$$

which gives us the form vector

$$\hat{\boldsymbol{c}}_{\text{kicked}} = \begin{array}{c} \begin{array}{ccccccccc} \text{\#t} & \text{tɔ} & \text{ɔk} & \text{k\#} & \text{kt} & \text{t\#} & \text{\#k} & \text{kɪ} & \text{ɪk} \end{array} \\ \left(\begin{array}{ccccccccc} -0.02 & -0.02 & -0.02 & 0.16 & 0.82 & 0.82 & 1.00 & 1.00 & 1.00 \end{array}\right) \end{array}. \tag{21}$$

The values in $\hat{\boldsymbol{c}}_{\text{kicked}}$ indicate the amount of semantic support that each biphone receives. Unsurprisingly, the biphones #k, kɪ, ɪk obtain very high values, suggesting that according to network predictions,

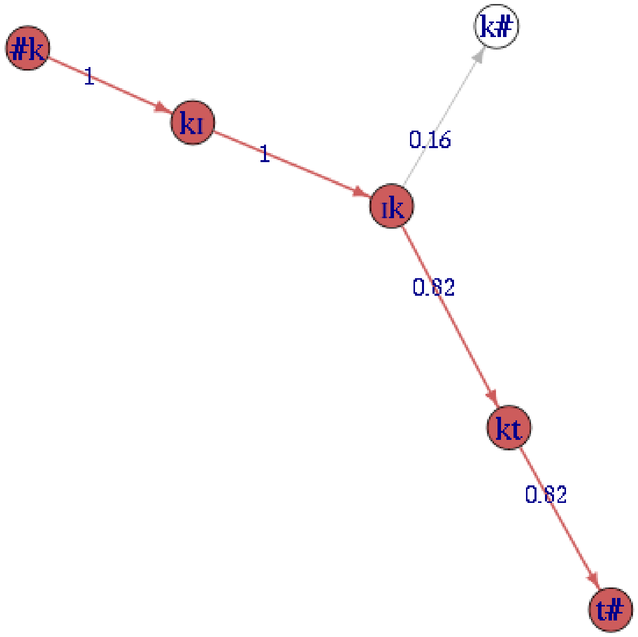

Figure 3: The biphone graph for the predicted form path of /kɪkt/, marked in red. By linking the highly supported cues together we obtain the predicted pronunciation form.

the word form for *kicked* should definitely contain these three biphones. Interestingly, the values for `kt` and `t#` are also high (0.82), and although for `k#` the support is low (0.16), it is not negative.

At this point, we know for each biphone how well it is supported by the semantics, but we don't have information about their order. However, as biphones (and triphones) contain implicit order information, they can be strung together when beginnings and ends match properly: `kt` and `t#` can be merged into `kt#`, but `kt` and `p#` cannot be joined. When all the biphones with positive support are linked together (the red path in Figure 3), we obtain the form /kɪkt/. As also shown in Figure 3, the form /kɪk/ is another possibility, but /kɪkt/ is a better candidate than /kɪk/ since it is better supported by semantics.

Why does the network prefer the /kt/ ending over /k/ for the past tense form? Table 4 presents the correlations between the semantic vectors of the two tenses, PRESENT and PAST, and the weight

Table 4: The correlations of the semantic vectors of PRESENT and PAST with the weight vectors of k# and t# (k# and t# columns in the network $\boldsymbol{G}$).

	k#	t#
PRESENT	0.99	-0.14
PAST	-0.27	0.93

vectors of the two biphones, k# and t# (columns 4 and 6 in $\boldsymbol{G}$ respectively). The vector of PRESENT is highly correlated with k#, while the vector of PAST is highly correlated t#. In other words, with the mathematics of multivariate multiple regression, equivalent to applying the discriminative learning algorithm iteratively for an infinite number of learning events, the network has learned to associate the meaning of PRESENT with k#, and the meaning of PAST with t#. At the same time, the network has also learned to disassociate PRESENT with t# and PAST with k#, here correlations are negative. Importantly, the model was not informed about any inflectional rules, nor about stems and exponents. All it did was learn the connection weights between sublexical cues (biphones) and semantic features. Nevertheless, it captured the proportional analogy for this agglutinative example: the network shifts support to t# for the past tense, and to the k# for the present tense. For fusional morphology, the proportional analogies are, however, distributed over many more cues.

3. The discriminative lexicon model

The discriminative lexicon model (Baayen et al., 2019) brings together a set of interconnected networks (Figure 4) that together account for a range of lexical processing tasks. Networks are indicated by arrows, and cues and outcomes are indicated by gray boxes. When weights are estimated at the EoL, we need the matrix representations of all cues and all outcomes together. For incremental learning, pairs of vector representations are required, one for the cues and one for the outcomes. As the network algorithm is fixed, the performance of the model hinges completely on how forms and meanings are represented.

- **Auditory forms** can be represented by low-level acoustic cues directly extracted from the speech signal. Arnold et al. (2017) developed Feature Band Summary (FBS) features, which

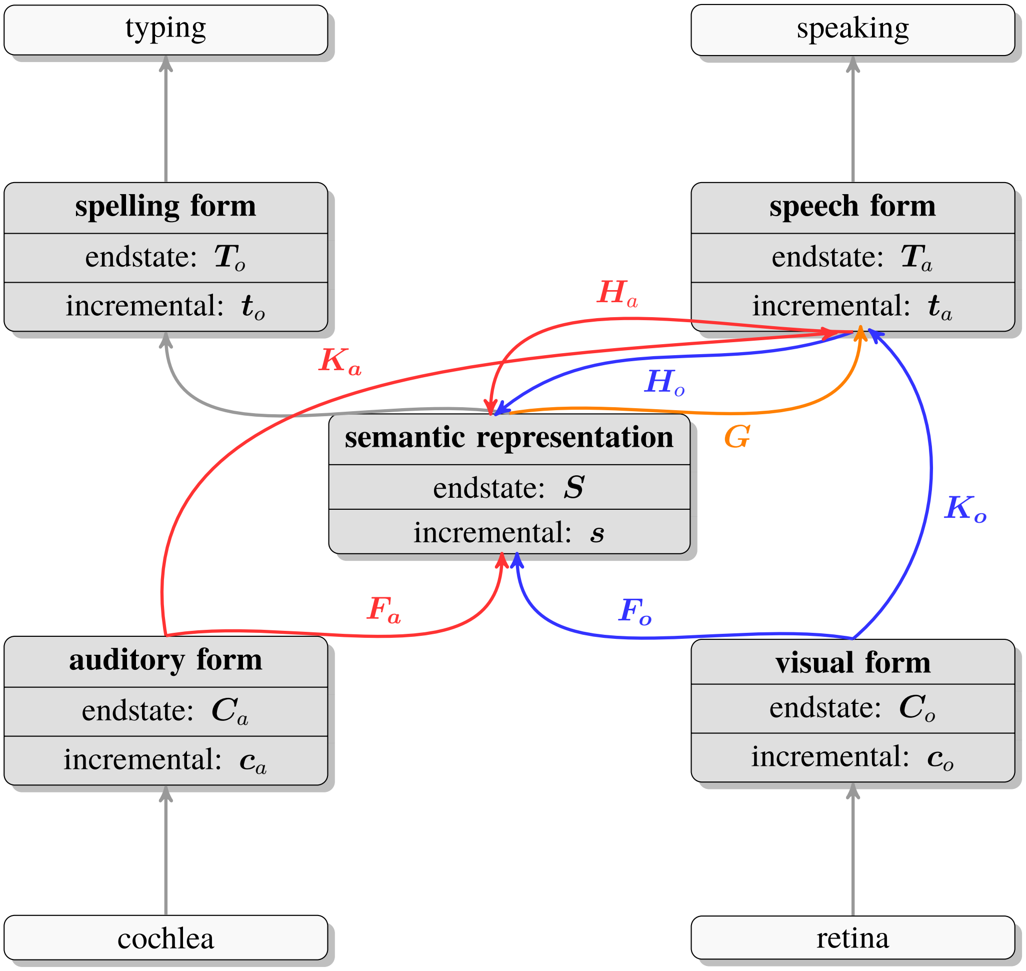
typing
speaking
spelling form
endstate: $\boldsymbol{T}_o$
incremental: $\boldsymbol{t}_o$
speech form
endstate: $\boldsymbol{T}_a$
incremental: $\boldsymbol{t}_a$
$\boldsymbol{H}_a$
$\boldsymbol{K}_a$
$\boldsymbol{H}_o$
$\boldsymbol{G}$
semantic representation
endstate: $\boldsymbol{S}$
incremental: $\boldsymbol{s}$
$\boldsymbol{K}_o$
$\boldsymbol{F}_a$
$\boldsymbol{F}_o$
auditory form
endstate: $\boldsymbol{C}_a$
incremental: $\boldsymbol{c}_a$
visual form
endstate: $\boldsymbol{C}_o$
incremental: $\boldsymbol{c}_o$
cochlea
retina

Figure 4: Overview of the discriminative lexicon. Input and output systems are presented in light gray, the vector representations characterizing the state of form and meaning subsystems are shown in dark gray. The vectors of individual words are brought together as the row vectors ($\boldsymbol{c}_o, \boldsymbol{c}_a, \boldsymbol{s}, \boldsymbol{t}_a$, and $\boldsymbol{t}_o$) of the matrices $\boldsymbol{C}_o, \boldsymbol{C}_a, \boldsymbol{S}, \boldsymbol{T}_a$, and $\boldsymbol{T}_o$. Arrows represent mappings between vectors. Mappings marked in red, $\boldsymbol{F_a}$, $\boldsymbol{K_a}$, and $\boldsymbol{H_a}$, are used to model auditory comprehension, and mappings marked in blue, $\boldsymbol{F_o}$, $\boldsymbol{K_o}$, $\boldsymbol{H_o}$, are for visual comprehension. The mapping $\boldsymbol{G}$ (orange) is for production.

are discrete features that summarize energy distributions at different frequency bands. Shafaei-Bajestan et al. (2020) further developed these features into real-valued vectors with an improved temporal resolution. When no audio files are available, phone n-grams can be used as idealized representations of auditory input.

- **Visual forms** can be represented at different levels of granularity. At a very low level, Linke et al. (2017) made use of histograms of oriented gradients (HOG) features (Dalal and Triggs, 2005) to represent English four-letter words. However, most actual models using NDL or LDL for studying visual word recognition have made use of more abstract, higher level n-gram representations, analogous to the n-phone representations illustrated in (17).

- **Speech forms** have thus far been represented by triphones or quadrophones. Envisioned future representations can be sequences of parameter values that drive aspects of actual pronunciation, such as the articulatory position of the tongue tip over time, or the fundamental frequency over time (Sering et al., 2019).

- **Spelling forms** can be represented by letter n-grams, or by kinesthetic features associated with finger movements during typing or writing. Up till now, no actual modeling of spelling has been done.

- **Semantic representations** can be formalized in many ways. The simplest is one-hot encoding, as implemented in NDL. Alternatively, real-valued vector representations can be used, as implemented in LDL. Such semantic vectors (aka word embeddings) can be derived from corpora with any of a wide range of algorithms, such as LSA (Landauer and Dumais, 1997), HAL (Lund and Burgess, 1996), HiDEx (Shaoul and Westbury, 2010), Word2vec (Mikolov et al., 2013) among many others. There are many ways in which the semantic vectors for complex words can be set up. Baayen et al. (2011) implemented a full-decomposition approach in which complex words never have their own lexomes. By contrast, Milin et al. (2017b) took a full-listing approach, in which complex words were treated at the semantic level in exactly the same way as monomorphemic words. LDL honors the distinction between inflection and word formation. The semantic vectors of inflected words are obtained by adding the inflectional vectors of the pertinent lexomes for the word's lemma and its inflectional functions. Derived words, on the other hand, are treated as simple words in that they receive their own semantic vectors. At the same time, derivational lexomes such as negation (*un-* or *-less*) and abstraction (*-ness*) are also brought into the model with their own semantic vectors.

In Figure 4, the red arrows represent networks that are involved in auditory comprehension. One network, $\boldsymbol{F_a}$, maps auditory input directly onto semantic representations. Alternatively, auditory input can first be mapped onto speech forms (via network $\boldsymbol{K_a}$), and then onward to the semantic representations (via network $\boldsymbol{H_a}$). The design of this indirect route is in line with the Motor Theory of Speech Perception, according to which articulatory gestures are the object of speech perception (Liberman and Mattingly, 1985). With respect to visual comprehension (blue arrows), Baayen et al. (2019) also studied a dual-route set-up. Phonological forms play a critical role in reading: actually, we can hear our 'inner voice' even in silent reading (Perrone-Bertolotti et al., 2012). The direct route is implemented by the network $\boldsymbol{F_o}$ that maps visual forms onto semantic representations. The indirect route requires two networks. $\boldsymbol{K_o}$ maps visual input onto speech forms, which are in turn mapped onto semantic representations by $\boldsymbol{H_o}$. Baayen et al. (2019) observed for their LDL model that comprehension accuracy was higher with the indirect route than with the direct route. This advantage of the indirect route, however, does not hold for Mandarin Chinese. The writing system of Chinese is logographic, not alphabetic, and a number of different characters can share the same pronunciation. The widespread homophony of Chinese has as a consequence that the visual cues provided by HOG features derived from Chinese characters are much more discriminative for meaning than the corresponding phonological representations. Simulations show that for visual word recognition in Mandarin, the indirect route performs worse than the direct route. Finally, the network $\boldsymbol{G}$ (orange arrow) represents the mapping involved in speech production, which takes semantic representations as input and maps these onto representations supposed to drive articulation.

4. Assessing model performance

The discriminative lexicon is a theory of the mental lexicon. Its validity and usefulness hinge on the accuracy of its predictions. Model performance can be evaluated in several ways. Internal validation seeks to establish whether the model is adequately producing words' forms and understanding their meanings. External validation pits the predictions of the model against experimental measures of lexical processing. External validation makes sense only for models that have passed the test of internal validation. In what follows, we first introduce the measures and methods on which validation depends (Section 4.1). Next, the results of internal validation are presented in Section 4.2. For external validation, a further distinction is made between item-level and system-level validation. The former takes place at the level of the processing properties of individual words (Section 4.3), whereas the latter is carried out at higher levels of abstraction, comparing system-wide properties of human processing with system-wide properties of the model (Section 4.4).

4.1 Measures and methods of evaluation

4.1.1 Internal validation: evaluating model accuracy

Evaluation of model performances is straightforward for NDL. For a given set of cues, we first calculate the activation for each outcome. The activation a_j of a particular outcome j is the predicted support that this outcome receives from its corresponding cues. It is defined as the sum of weights on the connections from all the pertinent cues i in the input to outcome j:

$$a_j = \sum_i w_{ij}. \tag{22}$$

Applying (22) to all outcomes, we obtain a vector $\boldsymbol{a}$ which contains the activation values of all the outcomes. The outcome that receives the highest activation is the model's prediction. For example, in a word recognition task, the outcome with the highest activation is considered as the recognized meaning.

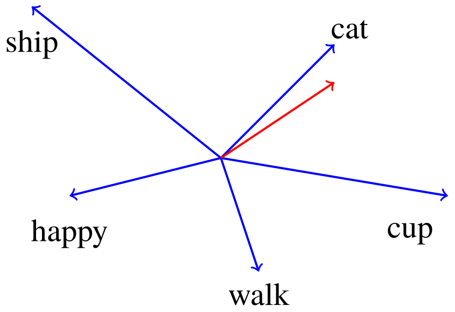

Figure 5: A small lexicon with five words, the semantic vectors of which are marked in blue. The red arrow represents one predicted semantic vector. Since the red vector is the closest to the vector of *cat*, the predicted meaning is thus CAT.

For LDL, assessing model performance in comprehension is less straightforward, since the meaning of a given word is represented by a vector of real numbers, instead of by individual outcomes. We therefore need to evaluate the predicted values of all outcomes simultaneously. For this we calculate the correlations of the predicted semantic vector with all the semantic vectors in the lexicon. Here, the correlation quantifies the similarity between two semantic vectors, with high correlations indicating high similarities. The recognized meaning is defined as the one with the highest correlation. By way of example, Figure 5 represents the semantic vectors $\boldsymbol{s}$ of five words in a toy lexicon in blue. The red arrow represents the predicted semantic vector $\hat{\boldsymbol{s}}_j$ for word j. Given that the predicted vector is the closest to the vector of *cat*, and hence has the highest correlation r, CAT is the recognized meaning.

As to production, for a given meaning $\boldsymbol{s}_i$, the model outputs a predicted form vector $\boldsymbol{c}_i$ specifying the amount of semantic support that each cue obtains (for an example, see 21). Several algorithms are available that take this form vector and generate the pronunciation form (as an ordered sequence of n-phones). (For larger datasets, the algorithms in the **JudiLing** package for Julia (Luo et al., 2021) are recommended.) Production errors occur when a predicted form is not identical to the targeted form.

4.1.2 External validation: model-based predictors

Both the activations a_k and the correlations r_k have been used successfully to predict for word k measures of lexical processing such as reaction times. Two other NDL-specific measures that are also predictive are activation diversity `ActDiv` and network prior `Prior`. The former is given by the L1-norm of the activation vector for input word k, which is the sum of the absolute values in its activation vector $\boldsymbol{a}_k$:

$$\text{ActDiv}_k = \sum |\boldsymbol{a}_k|. \tag{23}$$

This measure assesses the uncertainty about what the targeted meaning is. The network prior is calculated from the weight matrix. For a given outcome O_j, its prior $\texttt{Prior}_j$ is the L1-norm of column vector j in the weight matrix, which is the sum of the absolute values of the weights of all cues to this outcome:

$$\text{Prior}_j = \sum |w_{ij}| \tag{24}$$

This prior measures the extent to which a word is entrenched in the model's network, it is named a prior as its value is independent of the actual input to the network. Several other measures can be derived from NDL networks (Milin et al., 2017a,b; Divjak et al., 2020), of which specifically the L1-norm (prior) of corpus-based semantic vectors obtained with NDL has been found to be a strong predictor of reaction times in visual lexical decision.

Within the framework of LDL, further measures for comprehension can be derived for assessing where a word's predicted semantic vector $\hat{\boldsymbol{s}}$ is in semantic space. For instance, one can calculate how close $\hat{\boldsymbol{s}}$ is to its target semantic vector, and one can likewise derive measures on how many semantic vectors are in close proximity to $\hat{\boldsymbol{s}}$. For production, the amount of support from the semantics for the cues can be used to predict acoustic durations of the corresponding phones.

4.2 Internal validation results

Internal validation assesses model accuracy, not only for the training data, but also for unseen data. What counts as unseen depends on the task. A model cannot be expected to produce a monomorphemic English word it has never encountered before. It should however be able to predict unseen inflected variants of a known base word. For auditory comprehension, the model should also be able to understand novel audio tokens of previously heard words.

Baayen et al. (2019) reported good prediction accuracy for English inflectional and derivational morphology in reading. For auditory comprehension of English spoken words, Shafaei-Bajestan et al. (2020) fitted the model to some 130,000 word tokens from a corpus of spontaneous conversational English speech. They observed an accuracy of 25% on the training data, and an accuracy of 16% for under cross-validation. To put this in perspective, Mozilla Deep Speech (Hannun et al., 2014), a state-of-the-art machine learning technique, achieved an accuracy of only 6% on this task. Chuang et al. (2020b) studied performance for Estonian nouns, which inflect for number (singular, plural) and case (14 cases for each number). For 232 nouns from 26 inflectional classes, all 28 inflected variants were collected. Comprehension accuracy on training data was at 99.2%, in a model using orthographic letter triplets as cues and simulated semantic vectors constructed as described in equations (15–16). For production, accuracy on training data was at 99%. When trained on only those inflected forms that were attested in a corpus, and given the task to predict the remaining unseen inflected forms, comprehension accuracy remained high (97.5%), but production accuracy suffered (69.5%). The technical problem here is that letter trigrams that have not been encountered in the training data are not available for the production of unseen forms. By training on more comprehensive sets of words, this error rate is expected to decrease. These three studies all made use of LDL evaluated at EoL (LDL-EoL).

4.3 External validation results

4.3.1 Modeling reaction times

Lexical decision latencies (RTs) are one measure of lexical processing that have been used in external validation studies. Baayen et al. (2011) took NDL-EoL activations (22) and investigated their predictivity for visual RTs in the English Lexicon Project (Balota et al., 2004). The effect sizes of a range of predictors (such as word frequency, morphological family size, number of orthographic neighbors) were remarkably similar for regression models fitted to the observed RTs and the same regression models fitted to the activations. Unprimed auditory lexical decision latencies for auditory nonwords were modeled by Chuang et al. (2020c) with LDL-EoL. Their model was first trained on the real words in the MALD database (Tucker et al., 2019a), and then used to predict RTs to auditory nonwords. Correlation measures evaluating nonwords' predicted semantic vectors with neighboring real word vectors provided strong predictors for nonword RTs. Another measure evaluates how different a listener would have expressed the nonword meaning. The greater the distance between what listeners heard and would have said themselves, the faster a nonword could be rejected.

Milin et al. (2017b) studied masked primed lexical decision latencies for a dataset constructed to test the hypothesis that reading involves an obligatory early stage of morpho-orthographic segmentation (Rastle et al., 2004). NDL measures (obtained with incremenal learning) that they reported to be predictive were the `ActDiv` (23) and the `Prior` (24) (see also Baayen et al., 2016a). They observed a strong effect of the `Prior`, in interaction with `ActDiv`. A larger `prior` afforded shorter RTs. Thus, if a word is more probable a priori, before having seen any input, it is more likely to elicit a fast response. Conversely, a greater `ActDiv` predicted longer reaction times: greater uncertainty about what the intended target word is slowed participants. Interestingly, a regression model using classical predictors provided a less good fit than a regression model in which the classical predictors were replaced by model-based predictors.

Baayen and Smolka (2020) modeled morphological processing as gauged by an overt priming task. Primes were morphologically complex German verbs that preceded their base verbs. An incremental NDL model was trained on some 18,000 lemmas from the CELEX database, with letter triplets as cues. Measures predicting primed RTs were the activation of the target word by the cues in the prime word, and the activation of the target word by the cues of the target word itself. Interestingly, the prime-to-target activation measure perfectly captured the empirical priming pattern, indicating that the sub-lexical distributional properties of the German lexicon are at issue.

4.3.2 Modeling acoustic durations

A second measure of lexical processing modeled with LDL-EoL is acoustic duration. Baayen et al. (2019) investigated the acoustic duration of English stem-final segments such as /d/ in *blending, blends*, and *blended*. They observed that the amount of support the triphone centered on the /d/ (/ndɪ, ndz, ndɪ/) receives from the semantics of its carrier word predicts its acoustic duration. Segments with no support from the semantics have zero duration, segments with modest support have intermediate durations, and well-supported segments have long durations.

Tomaschek et al. (2019) used incremental NDL to model the duration of word-final [s] in English. Plag et al. (2017) had previously observed that stem-final, non-morphemic [s] is longer than all morphemic [s] homophones (plural, genitive, genitive plural, 3rd person singular, and the cliticized forms of *has* and *is*), and that within the set of morphemic [s] homophones, the plural and 3rd person singular exponents have durations that are longer than those of the clitics. To model [s] durations, Tomaschek et al. (2019) extracted the two preceding and following words. As cues for learning, they considered the five words in the resulting context window, together with all phone pairs in this window. As outcomes, they used one-hot encoded semantic vectors for the words and the different inflectional functions, similar to the decompositional approach of semantic representations in Baayen et al. (2011). The duration of [s] increased nearly linearly with the `Prior` (24) of a word's inflectional function, i.e., the L1-norm of the semantic vector of inflectional lexomes such as PLURAL or GENITIVE. As an inflectional/lexical function is more entrenched in the network, it is pronounced more confidently and with longer duration. This longer duration in production mirrors the effect of the prior in comprehension, where shorter reaction times are seen as a function of a greater prior.

Tomaschek et al. (2019) observed two further measures to be predictive: the activation (22) of the inflectional function and the activation diversity (23) of the set of inflectional functions. A greater activation afforded longer acoustic durations, whereas a greater activation diversity gave rise to shorter duration. Just as a greater activation diversity in comprehension gives rise to longer RTs, in production it gives rise to shorter durations. By contrast, a greater activation support reduces uncertainty about what to say, and thereby affords longer duration. Similar results for stem vowel duration in English are reported by Tucker et al. (2019b).

The above-mentioned study by Chuang et al. (2020c) on the semantics of auditory nonwords not only considered auditory lexical RTs as the response variable, but also the acoustic duration of the nonwords as produced by the MALD's speaker. They observed the same measures that were predictive for the RTs to be predictive for word duration as well.

4.3.3 Modeling the time-course of learning

When information is available about the order in which words are encountered, the Rescorla-Wagner or Widrow-Hoff incremental learning rules can be used to trace how learning develops over time.

A first example of incremental lexical learning comes from the field of animal learning. Grainger et al. (2012) trained baboons to discriminate between real English words and nonwords. Hannagan et al. (2014) modeled baboon performance over time with a deep convolutional network. Analysis of the deep learning network's internal representations suggested that it had developed selective sensitivity to position-specific letters, bigrams and trigrams. The authors concluded that baboons' reading abilities are mediated by a similar hierarchical processing system as exist in the ventral pathway of the human brain. However, statistical examination of baboon performance (Linke et al., 2017) revealed strong effects of whole words, but weak and inconsistent effects of letters, letter bigrams, and letter trigrams. Therefore, Linke et al. (2017) used incremental NDL to model baboon learning. Their network had two outcome units, one for a yes response and one for a no response. On its input layer, the model made use of roughly 14,500 discrete low-level visual features constructed from HOG features. The model for a given baboon was exposed to the words and nonwords in exactly the same order as received by that baboon. At each successive learning event, the weights in the network were adjusted twice. The first adjustment was driven by the baboons own response (word or nonword). Next, the model was updated by the feedback it received from the experimental apparatus (correct or incorrect response). Model performance over time resembled baboon behavior more closely than was the case for the deep learning model, with ups and downs in baboon accuracy being well mirrored by similar ups and downs in model accuracy.

As a second example of incremental learning, we consider the modeling of RTs in the British Lexicon Project (BLP, Keuleers et al., 2012). Unlike baboons exposed for the first time to letter strings, the participants in the BLP came into the experiment with detailed knowledge of English. To model this prior knowledge, we constructed an NDL network predicting lexomes from letter trigrams. This network was trained incrementally on nearly 5 million sentences of the written part of the British National Corpus (BNC Burnard, 1995), resulting in a network of some 15,000 trigram cues and some 30,000 lexome outcomes. For each word that occurs in both the BNC and the BLP, we calculated the `ActDiv` (23) and `Prior` (24).

A baseline is provided by a model using only 'static' lexical predictors. For visual lexical decision, the most studied static predictors are frequency of occurrence, length in letters, and orthographic neighborhood density (N-count). The first row of Table 5 lists two goodness of fit statistics for a baseline model with these three predictors, the maximum likelihood (ML) and Aikaike's Information Criterion (AIC). Smaller values of these statistics indicate a better fit. The second row of Table 5 shows that when the N-count measure is replaced by the static predictors `Prior` and `ActDiv` derived from the BNC, model fit is slightly less good, even though `ActDiv` and `Prior` are both evaluated as significant.

The third row of this table lists the results obtained with a dynamically updated network. In order to model word and nonword decisions, we extended the network's outcomes with two additional lexomes, one for a yes decision and one for a no decision. We presented the nearly 14,000 words in the BLP available for participant 1 to the network in exactly the same order in which they were presented. For each trial, we first obtained the network's predictions for `ActDiv` and `Prior`. We then updated the weights of the network as follows. If the participant had provided a yes response, the connections from the word's trigrams to the word's lexome and the yes lexome were strengthened, and the connections from these trigrams to all other lexomes (including the no lexome) were weakened, according to the Rescorla-Wagner learning rule (2). For no-responses, connections to the no lexome were strengthened and those to all other word lexomes and the yes lexome were weakened. This makes it possible to assess the amount of support for a yes response that builds up

Table 5: Model fit for generalized additive models fitted to the visual lexical decision latencies of participant 1 in the British Lexicon Project. Lower ML and AIC values indicate better fits. Predictors that are updated after every trial are shown in bold.

	ML	AIC	predictors
classical static	-1759	-3873	frequency, length, N-count
NDL static	-1737	-3825	frequency, length, ActDiv, Prior
NDL dynamic	-1817	-4004	frequency, length, **activation yes response, ActDiv, Prior**

as the participant is encountering words and nonwords as the experiment unfolds.

The final row of Table 5 lists ML and AIC values for the model using the three dynamic predictors `ActDiv`, `Prior`, and the activation of the `yes` lexome. This dynamic model provides a substantially improved goodness of fit. The improvement in goodness of fit obtained by updating the NDL measures after every trial clarifies that the discriminative learning approach is powerful enough to capture aspects of the ongoing process of re-calibration of the lexicon that unfolds as we use our language. For similar results for L1 acquisition, see Ramscar et al. (2013b) and for phonetic learning, Nixon (2020).

4.4 Modeling systemic properties of lexicons

Instead of evaluating the model on how well it predicts RTs for individual words or acoustic durations for individual words or segments, we can ask whether the model properly mirrors higher-level properties of the language system. More specifically, we can ask whether qualitative differences in lexical processing can be accounted for as resulting from differences in the distributional properties of cues and outcomes. If discriminative learning is on the right track, these differences should emerge straightforwardly from discriminative learning.

First consider the systematic differences in lexical processing that have been observed for English and Hebrew. For English, transposing letters (e.g., writing *perhaps* as *pehraps*) is relatively harmless. Readers can reconstruct the intended meaning, and when reading for content, typos of this kind often escape notice. However, in Hebrew, letter transpositions severely disrupt reading (Frost, 2012). To understand why, Baayen (2012) applied incremental NDL to English and Hebrew datasets of comparable size, using letter pairs as cues. Hebrew emerged as using a smaller set of bigram cues, while using these cues more intensively, such that Hebrew bigram cues tended to have stronger weights to lexome outcomes compared to English. Interestingly, the disruption caused by transposing letters can be estimated by calculating the extent to which a word's activation decreases when a pair of letters is transposed. Median disruption in Hebrew was almost 6.6 times that for English, which explains why Hebrew is more vulnerable to letter transpositions than English. Other systemic differences between Hebrew and English reading, such as the relative immunity of loanwords in Hebrew to letter transpositions, and the absence of facilitation from form priming in Hebrew, also follow straightforwardly from discriminative learning. Thus, discriminative learning helps clarify that indeed, as hypothesized by Frost (2012), the very different distributional properties of the Hebrew and English lexicons lie at the heart of the qualitative differences observed experimentally for lexical processing in these languages.

A second example of a system property that discriminative learning handles well is a double dissociation observed for English. Under impairment, English speakers with memory loss tend to have greater difficulties with irregular verbs, whereas speakers with phonological impairment tend to have more problems with regular verbs. Joanisse and Seidenberg (1999) used a connectionist network to model this dissociation, and observed that they had to selectively add noise to the semantic units of their model in order to get it to work properly. Interestingly, Joanisse and Seidenberg (1999) used one-hot encoding for their semantic representations, under the assumption that regular and irregular verbs do not differ systematically in meaning. However, irregular verbs have denser semantic neighborhoods than do regular verbs (Baayen and Moscoso del Prado Martín, 2005). Heitmeier and Baayen (2021) therefore used LDL-EoL to model the double dissociation of regularity by impairment, as LDL makes it possible to work with corpus-based semantic vectors that reflect the differences in semantic density of regular and irregular verbs. Their simulation studies show that it is indeed the greater semantic density of English irregular verbs that renders them more fragile under semantic impairment.

A third example of using discriminative learning at the system level concerns second and third language acquisition. Chuang et al. (2020a) carried out a series of simulations with translation equivalents from German, English, Mandarin, and Dutch, using both incremental LDL and LDL-EoL. As expected, the onset of L2 learning has a substantial influence on learning, as does the amount of L2 input. Another factor that emerged as important is the number of homophones in a language. Within the L1, homophones can be handled fairly well. However, when additional languages are learned, the homophones become more fragile and begin to suffer from intrusion, specifically if the translation equivalents of the other languages are not themselves homophonous. The fragility of homophones fits well with the problems that L2 learners have with L2 words that have multiple senses that do not overlap with the senses of their L1 translation equivalents. Take English *cut* for example. One can *cut* with an axe, a knife, or scissors, whereas in Dutch and Mandarin the different cutting actions have to be expressed by three different verbs: *hakken, knippen*, and *snijden* in Dutch and *kǎn, qiē*, and *jiǎn* in Mandarin. L1 English speakers, when learning Dutch or Mandarin as L2, will thus have to learn to make more fine-grained decisions as to which verb to use as translation equivalent for English *cut*. From this perspective, *cut* actually has three meanings, and hence is a homophone. It straightforwardly follows that LDL predicts they are inevitably difficult for non-native speakers to learn.

5. General considerations

In computational linguistics, deep learning networks are the state-of-the-art. NDL/LDL contrasts with deep learning in several ways. First, there are no hidden layers. Second, the mathematics of multivariate multiple regression guarantees interpretational transparency. For the endstate of learning, convergence is to the global optimum. Third, LDL works well in sparse high dimensional spaces, whereas deep learning typically works with denser spaces of much lower dimensionality. Since discriminative networks typically have many thousands of units, they implement ‘wide learning’.

Importantly, dimension reduction can lead to considerable reduction in LDL’s prediction accuracy. For low-dimensional compressed spaces, deep learning is required, as mappings between compressed low-dimensional spaces typically are non-linear. In LDL, it is advisable to match the dimensionality of form vectors with that of the semantic vectors, for the mathematics to work best. As form matrices using binary coding to specify the presence or absence of n-grams or n-phones are very sparse, networks mapping form onto meaning can be simplified by removing connections with weights very close to zero (see, e.g. Arnold et al., 2017; Milin et al., 2017b), with hardly any loss of accuracy.

Fourth, NDL and LDL hardly have any hyperparameters. For incremental learning, the learning rate is a free hyperparameter. For form vectors based on units such as triphones or trigrams, the dimensionality of the form space is determined by the number of different triphones in the data, and is not a free hyperparameter. And since the dimension of semantic vectors should match that of the form vectors, this is not a free hyperparameter either. Thus, how the model performs is determined by, firstly, the way form and meaning vectors are constructed (see Milin et al., 2017b, for discussion of the choice of representation for solving nonlinear classification with NDL), and secondly, by the distributional properties of the data. From this perspective, NDL/LDL provides a statistical tool that is driven almost exclusively by the data.

Breiman (2001) distinguished between two modeling cultures, the data modeling culture of statistics, and the algorithmic modeling culture of machine learning. The goal of algorithmic modeling is to obtain precise predictions, and if the model remains a black box or is theoretically uninterpretable (as is often the case for deep learning models as well as with recursive partitioning models) this is not an issue. The goal of data modeling, by contrast, is to obtain a statistical model that can generate the observed data. The resulting model provides the analyst with insight into what could be the mechanisms that give rise to the data. Although LDL can be seen as the simplest possible way of implementing machine learning for lexical learning, it is in spirit much closer to statistical data analysis. It is designed to assist the analyst to better understand the relations between sublexical distributional properties and text-derived distributional semantic properties. Because LDL implements *linear* multivariate multiple regression, it inherits from the linear model the limitation that nonlinear functional relations (as addressed by, e.g., the generalized additive model, Wood, 2017) cannot be discovered. Here, there is room for considerable improvement.

An important property of the present formalization of Word and Paradigm Morphology is that morphological processing is not construed as involving parallel operations on symbols of form and meaning. Figuring out the proper forms in production, or the proper meanings in comprehension, is left to the network. Importantly, the network dynamically creates form and meaning representations, rather than retrieving them from some static lexical repository. Thus, NDL and LDL move away from classical compositional models in which stems and exponents have to be parsed out for comprehension or combined for production. However, the present approach is analytical in the sense that semantic vectors for inflected words are obtained by summing the semantic vectors of pertinent lexomes.

It has been argued that phonemes, stems, and exponents have corresponding specialized neural areas in the brain (Bozic et al., 2010; Cibelli et al., 2015). Since NDL and LDL are statistical models of lexical processing, they make no claims about the topological organization of discrimination networks in neural tissue. However, it is mathematically possible to impose topological organization on form and semantic units in a 2-dimensional plane. As shown in Baayen et al. (2018) and Shafaei-Bajestan et al. (2020), phone-like and morph-like clusters can emerge in such 2-D maps. (As shown by Heitmeier and Baayen (2021), such maps make it possible to implement topologically localized lesions in simulation studies of aphasia.) Thus, NDL and LDL are not necessarily at odds with experimental evidence that is usually understood as supporting the neural reality of phonemes and morphemes. Instead of taking phonemes and morphemes to be the atomic units of a linguistic calculus, they can be understood as emergent properties of discriminative learning networks. Crucially, these low-level statistical networks provide much more precise predictions for the fine details of lexical processing than can be obtained with high-level abstract symbolic systems (for a discussion of NDL and statistical learning in language acquisition, see Baayen et al., 2016b). However, since high-level symbolic descriptions are part of the cultural embedding of language, and have uncontestable pedagogical value, a full model of the mental lexicon will have to comprise not only low-level discriminative learning, but also high-level explicit symbolic learning that is grafted on top of low-level implicit learning.

Vectors and matrix multiplication

In what follows, we provide a brief conceptual introduction to vectors and matrices, using as example shopping lists and total expenses depending on supermarket. Consider Jack, who wants to buy two apples and three pairs, and Jill, who wants to buy four apples and two pairs. Their respective shopping lists can be represented as vectors, ordered sequences of numbers: $(2, 3)$ for Jack and $(4, 2)$ for Jill. We bring these two shopping list vectors together in a 2×2 matrix $\boldsymbol{C}$.

$$\boldsymbol{C} = \begin{array}{cc} & \begin{array}{cc}\text{apples} & \text{pears}\end{array} \\ \begin{array}{c}\text{Jack}\\ \text{Jill}\end{array} & \begin{pmatrix} 2 & 3 \\ 4 & 2 \end{pmatrix} \end{array}. \tag{25}$$

At the Edeka supermarket, apples cost €2 and pears €3. At the Rewe supermarket, they cost €3 and €2. We bundle this information in a second matrix, $\boldsymbol{F}$.

$$\boldsymbol{F} = \begin{array}{cc} & \begin{array}{cc}\text{edeka} & \text{rewe}\end{array} \\ \begin{array}{c}\text{apples}\\ \text{pears}\end{array} & \begin{pmatrix} 2 & 3 \\ 3 & 2 \end{pmatrix} \end{array}. \tag{26}$$

We can now calculate the total expenses at each of the supermarkets for both Jack and Jill. For Jack, buying 2 apples and 3 pears at Edeka will cost € $2 \times 2 + 3 \times 3 = 13$. This combination of weighting numbers by price, and adding, is formalized by matrix multiplication.

$$\boldsymbol{S} = \boldsymbol{CF} = \begin{array}{cc} & \begin{array}{cc}\text{apples} & \text{pears}\end{array} \\ \begin{array}{c}\text{Jack}\\ \text{Jill}\end{array} & \begin{pmatrix} \mathbf{2} & \mathbf{3} \\ 4 & 2 \end{pmatrix} \end{array} \cdot \begin{array}{cc} & \begin{array}{cc}\text{edeka} & \text{rewe}\end{array} \\ \begin{array}{c}\text{apples}\\ \text{pears}\end{array} & \begin{pmatrix} \mathbf{2} & 3 \\ \mathbf{3} & 2 \end{pmatrix} \end{array} = \begin{array}{cc} & \begin{array}{cc}\text{edeka} & \text{rewe}\end{array} \\ \begin{array}{c}\text{Jack}\\ \text{Jill}\end{array} & \begin{pmatrix} \mathbf{13} & 12 \\ 14 & 16 \end{pmatrix} \end{array}. \tag{27}$$

The numbers going into the calculation of the expenses for Jack at Edeka are bolded: the row vector of Jack's shopping list in $\boldsymbol{C}$ is paired with the column vector for Edeka in $\boldsymbol{F}$, the elements of these vectors are multiplied pairwise, and then summed. The other three total expenses listed in $\boldsymbol{S}$ are obtained in the same way. Jack is better off going to Rewe for his shopping, whereas Jill is better of going to Edeka.

When modeling comprehension with LDL, we know $\boldsymbol{C}$ and we also know $\boldsymbol{S}$, and we want to estimate $\boldsymbol{F}$. With $\boldsymbol{F}$ in hand, we can transform a 'shopping list' of triphones into a list of 'semantic expenses'.

Links to digital materials

An R package for discriminative learning, **WpmWithLdl**, is available at `http://www.sfs.uni-tuebingen.de/~hbaayen/publications/WpmWithLdl_1.0.tar.gz`. An optimized implementation of LDL in julia, **JudiLing**, is available at `https://megamindhenry.github.io/JudiLing.jl/stable/`.

Reference List

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. (2012). Learning from the bible: computational modelling of the costs of letter transpositions and letter exchanges in reading classical hebrew and modern english. *Lingue e linguaggio*, 11(2):123–146.

Baayen, R. H., Chuang, Y.-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. and Moscoso del Prado Martín, F. (2005). Semantic density and past-tense formation in three Germanic languages. *Language*, 81:666–698.

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.

Balota, D., Cortese, M., Sergent-Marshall, S., Spieler, D., and Yap, M. (2004). Visual word recognition for single-syllable words. *Journal of Experimental Psychology: General*, 133:283–316.

Blevins, J. P. (2016). *Word and paradigm morphology*. Oxford University Press.

Bozic, M., Tyler, L. K., Ives, D. T., Randall, B., and Marslen-Wilson, W. D. (2010). Bihemispheric foundations for human speech comprehension. *Proceedings of the National Academy of Sciences*, 107(40):17439–17444.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

Burnard, L. (1995). *Users guide for the British National Corpus.* British National Corpus consortium, Oxford university computing service.

Chuang, Y.-Y., Bell, M. J., Banke, I., and Baayen, R. H. (2020a). Bilingual and multilingual mental lexicon: a modeling study with linear discriminative learning. *Language Learning*, pages 1–73.

Chuang, Y.-Y., Loo, K., Blevins, J. P., and Baayen, R. H. (2020b). Estonian case inflection made simple. A case study in Word and Paradigm morphology with Linear Discriminative Learning. In Körtvélyessy, L. and Štekauer, P., editors, *Advances in Morphology*, pages 119–141. Cambridge University Press.

Chuang, Y.-Y., Vollmer, M.-L., Shafaei-Bajestan, E., Gahl, S., Hendrix, P., and Baayen, R. H. (2020c). The processing of nonword form and meaning in production and comprehension: A computational modeling approach using linear discriminative learning. *Behavior Research Methods*, pages 1–51.

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.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893.

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

Divjak, D., Milin, P., Ez-zizi, A., Józefowski, J., and Adam, C. (2020). What is learned from exposure: an error-driven approach to productivity in language. *Language, Cognition and Neuroscience*, pages 1–24.

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

Ellis, N. C. (2013). Second language acquisition. *The Routledge Handbook of Second Language Acquisition*, page 193.

Frost, R. (2012). Towards a universal model of reading. *Behavioral and Brain Sciences*, page in press.

Grainger, J., Dufau, S., Montant, M., Ziegler, J. C., and Fagot, J. (2012). Orthographic processing in baboons (papio papio). *Science*, 336(6078):245–248.

Hannagan, T., Ziegler, J. C., Dufau, S., Fagot, J., and Grainger, J. (2014). Deep learning of orthographic representations in baboons. *PLOS-one*, 9:e84843.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al. (2014). Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Heitmeier, M. and Baayen, R. H. (2021). Simulating phonological and semantic impairment of English tense inflection with Linear Discriminative Learning. *The Mental Lexicon*, in press. PsyArXiv.

Joanisse, M. F. and Seidenberg, M. S. (1999). Impairments in verb morphology after brain injury: a connectionist model. *Proceedings of the National Academy of Sciences*, 96:7592–7597.

Kapatsinski, V. (2018). *Changing minds changing tools: From learning theory to language acquisition to language change*. MIT Press.

Keuleers, E., Lacey, P., Rastle, K., and Brysbaert, M. (2012). The british lexicon project: Lexical decision data for 28,730 monosyllabic and disyllabic english words. *Behavior Research Methods*, 44(1):287–304.

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.

Liberman, A. and Mattingly, I. (1985). The motor theory of speech perception revised. *Cognition*, 21:1–36.

Linke, M., Broeker, F., Ramscar, M., and Baayen, R. H. (2017). Are baboons learning "orthographic" representations? probably not. *PLOS-ONE*, 12(8):e0183876.

Lund, K. and Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behaviour Research Methods, Instruments, and Computers*, 28(2):203–208.

Luo, X., Chuang, Y. Y., and Baayen, R. H. (2021). Judiling: an implementation in Julia of Linear Discriminative Learning algorithms for language modeling.

Matthews, P. H. (1974). *Morphology. An Introduction to the Theory of Word Structure*. Cambridge University Press, Cambridge.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Milin, P., Divjak, D., and Baayen, R. H. (2017a). A learning perspective on individual differences in skilled reading: Exploring and exploiting orthographic and semantic discrimination cues. *Journal of Experimental Psychology: Learning, Memory, and Cognition*.

Milin, P., Feldman, L. B., Ramscar, M., Hendrix, P., and Baayen, R. H. (2017b). 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. PsyArXiv.

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

Perrone-Bertolotti, M., Kujala, J., Vidal, J. R., Hamame, C. M., Ossandon, T., Bertrand, O., Minotti, L., Kahane, P., Jerbi, K., and Lachaux, J.-P. (2012). How silent is silent reading?

intracerebral evidence for top-down activation of temporal voice areas during reading. *Journal of Neuroscience*, 32(49):17554–17562.

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.

Ramscar, M., Dye, M., and Klein, J. (2013a). Children value informativity over logic in word learning. *Psychological Science*, 24(6):1017–1023.

Ramscar, M., Dye, M., and McCauley, S. M. (2013b). 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.

Rastle, K., Davis, M. H., and New, B. (2004). The broth in my brother's brothel: Morpho-orthographic segmentation in visual word recognition. *Psychonomic Bulletin & Review*, 11:1090–1098.

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. In Black, A. H. and Prokasy, W. F., editors, *Classical conditioning II: Current research and theory*, pages 64–99. Appleton Century Crofts, New York.

Sering, K., Stehwien, N., and Gao, Y. (2019). create_vtl_corpus: Synthesizing a speech corpus with vocaltractlab (version v1.0.0). Zenodo. `http://doi.org/10.5281/zenodo.2548895`.

Shafaei-Bajestan, E., Tari, M. M., and Baayen, R. H. (2020). LDL-AURIS: Error-driven learning in modeling spoken word recognition. arXiv.

Shaoul, C. and Westbury, C. (2010). Exploring lexical co-occurrence space using hidex. *Behavior Research Methods*, 42(2):393–413.

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*, 57(1):123–161.

Tucker, B. V., Brenner, D., Danielson, D. K., Kelley, M. C., Nenadić, F., and Sims, M. (2019a). The massive auditory lexical decision (mald) database. *Behavior research methods*, 51(3):1187–1204.

Tucker, B. V., Sims, M., and Baayen, R. H. (2019b). Opposing forces on acoustic duration. *PsyArXiv*.

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

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