EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Philosophische Fakultät
Seminar für Sprachwissenschaft

Naive discrimination learning

Peter Hendrix

Outline

- What is discrimination learning?
- How can we apply discrimination learning to language?
- Examples of applications:
  - Baayen et al. (2011)
  - Hendrix, Ramscar & Baayen (2015)

What is discrimination learning?

- “the process by which animals or people learn to make different responses to different stimuli”
- “learn from the discrepancy between what is expected to happen and what actually happens”

Rescorla-Wagner

- Association strength between cues and outcomes (Rescorla & Wagner, 1972):

  $$V_i^{t+1} = V_i^t + \Delta V_i^t$$

  where

  $$\Delta V_i^t = \begin{cases} 0 & \text{if } \textsc{absent}(C_i, t) \\ \alpha_i \beta_1 \left(\lambda - \sum_{\textsc{present}(C_j,\, t)} V_j\right) & \text{if } \textsc{present}(C_i, t)\ \&\ \textsc{present}(O, t) \\ \alpha_i \beta_2 \left(0 - \sum_{\textsc{present}(C_j,\, t)} V_j\right) & \text{if } \textsc{present}(C_i, t)\ \&\ \textsc{absent}(O, t) \end{cases}$$

- If a cue is reliable, it's connection strength will increase
- If a cue is unreliable, it's connection strength will decrease

Rescorla-Wagner

- RW equations represent learning over time
- Danks (2003) provided equilibrium equations

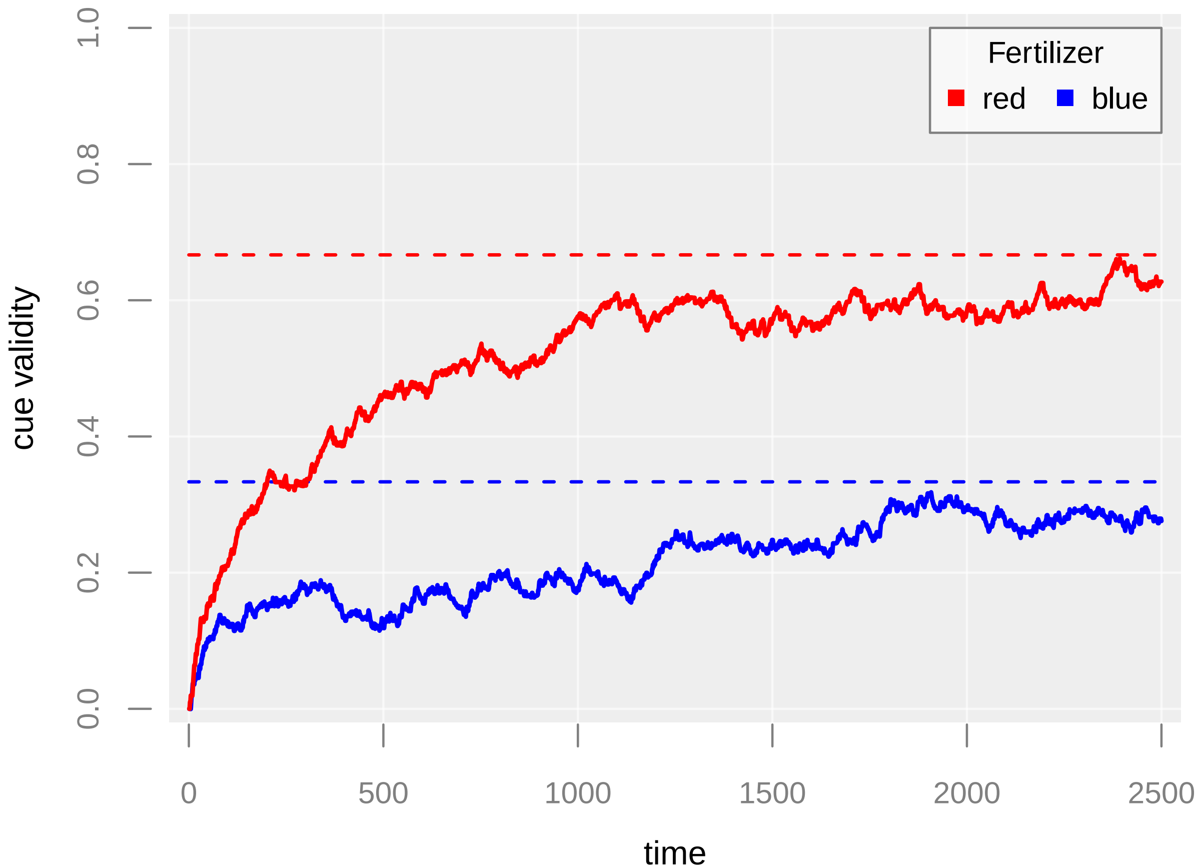
Example

- Fertilization of plants
- Pots
- Two types of fertilizer: red and blue

Example

Cues	Frequency	Outcome
red, blue, pot	5	yes
red, pot	10	yes
red, pot	5	no
blue, pot	5	yes
blue, pot	10	no
pot	5	no

Example


Fertilizer
red
blue
cue validity
0.0
0.2
0.4
0.6
0.8
1.0
0
500
1000
1500
2000
2500
time

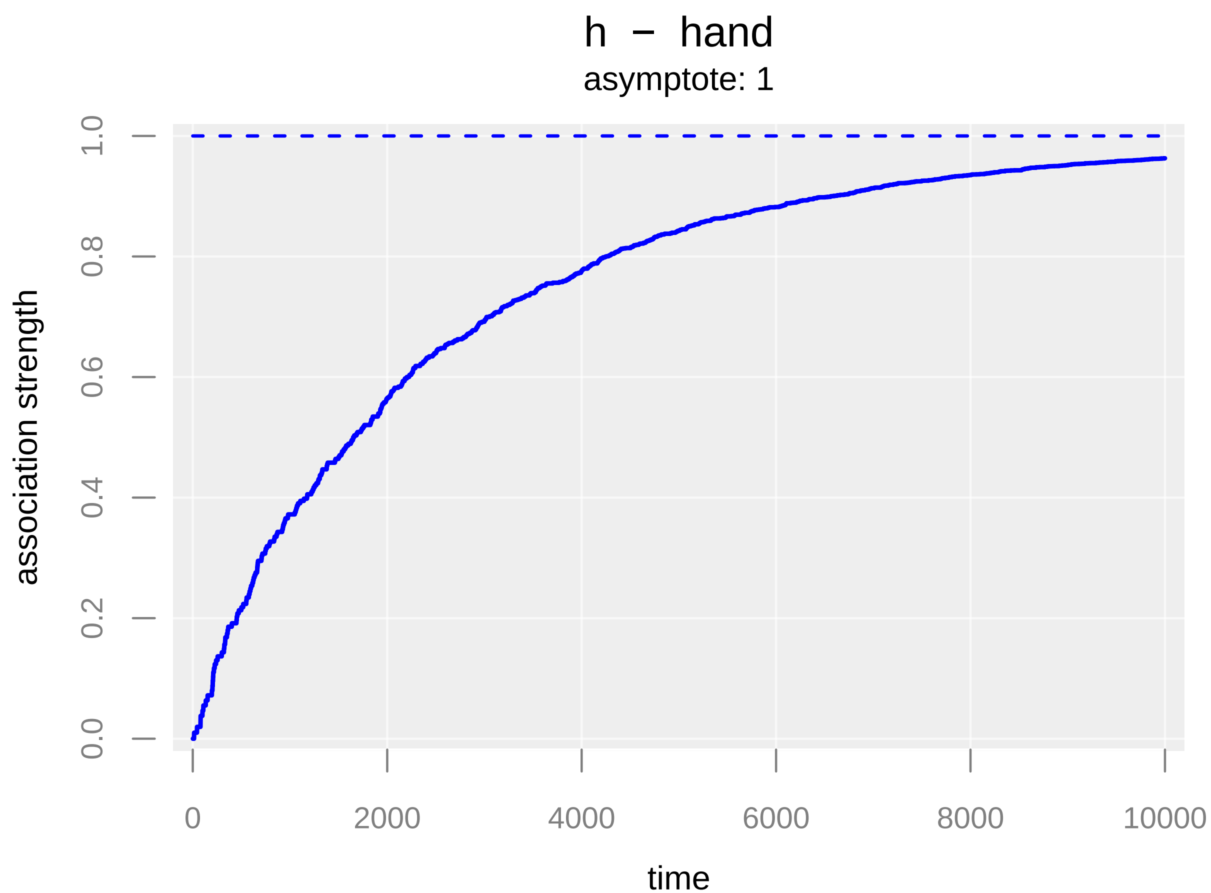
How can we apply discriminative learning to language?

- Model to learn semantics from orthography
- Cues: letters
- Outcomes: meanings
- RW model learns associations between letters and meanings

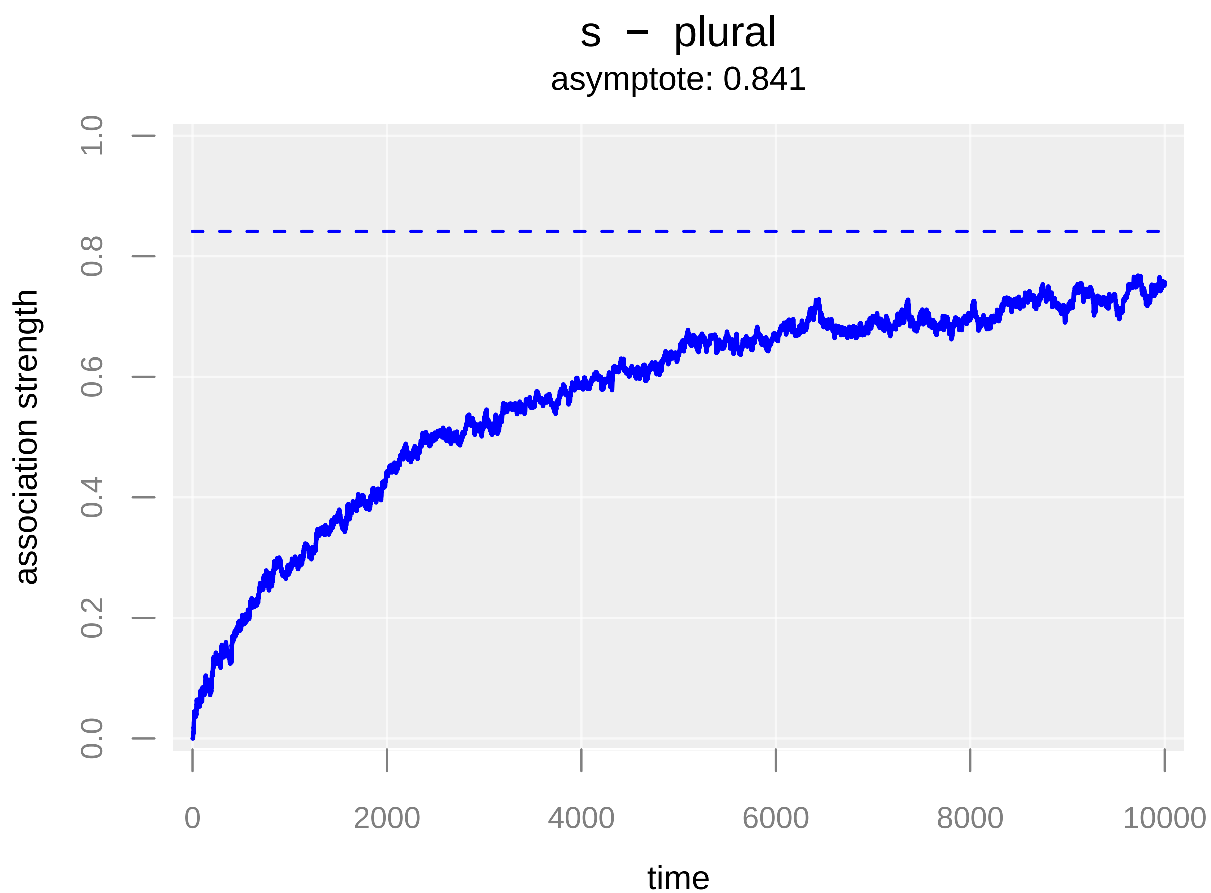
Example

Word	Cues	Frequency	Outcomes
hand	h, a, n, d	10	hand, NIL
hands	h, a, n, d, s	20	hand, plural
land	l, a, n, d	8	land, NIL
lands	l, a, n, d, s	3	land, plural
and	a, n, d	35	and, NIL
sad	s, a, d	18	sad, NIL
as	a, s	35	as, NIL
lad	l, a, d	102	lad, NIL
lads	l, a, d, s	54	lad, plural
lass	l, a, s, s	134	lass, NIL

Example

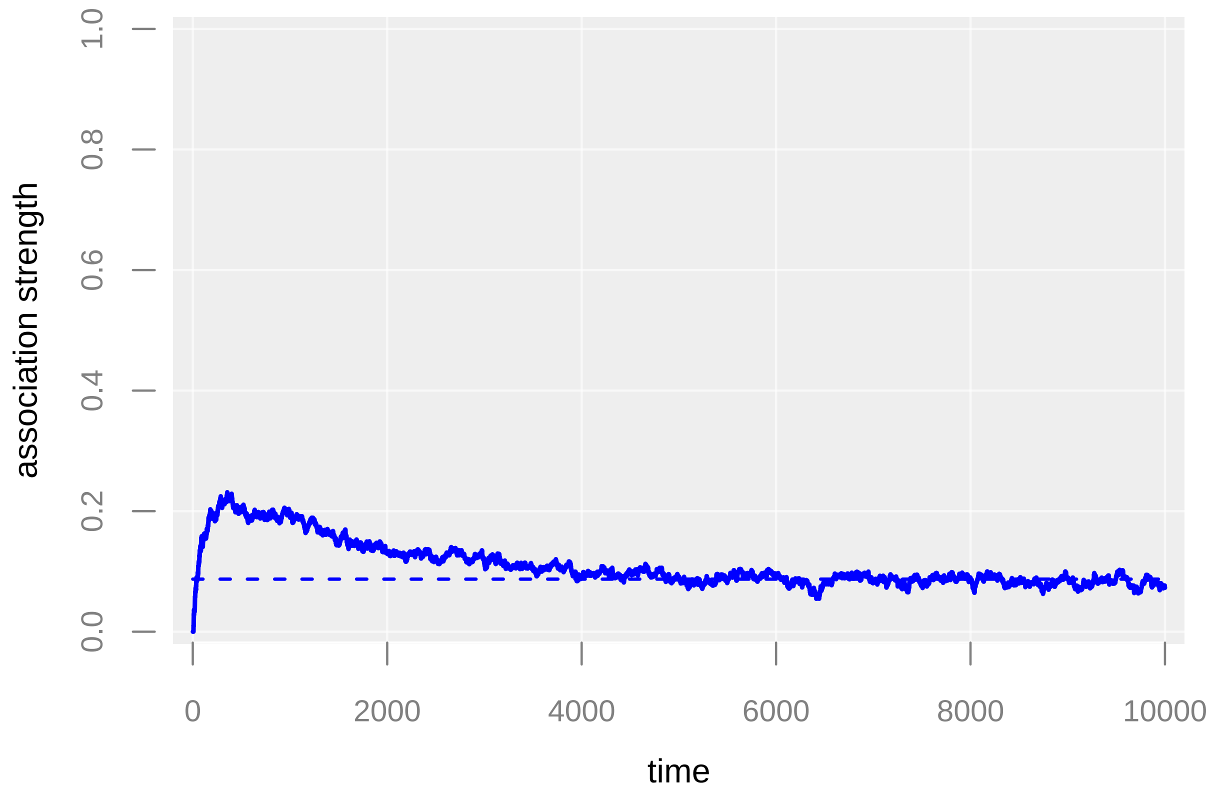

h – hand
asymptote: 1
association strength
0.0
0.2
0.4
0.6
0.8
1.0
0
2000
4000
6000
8000
10000
time

Example


s − plural
asymptote: 0.841
association strength
0.0
0.2
0.4
0.6
0.8
1.0
0
2000
4000
6000
8000
10000
time

Example

s – lass
asymptote: 0.087



Naive discrimination learning

- Apply the same idea on a larger scale
- Association strengths are learned separately for each outcome
- Implemented in the `ndl` package for R

Example

```
# Load data
load("data/lexicon.rda")
dim(lexicon)
# [1] 1293    4
lexicon$Word[1:20]
#  [1] "ace"   "age"   "aide"  "air"   "aisle" "ale"   "ant"   "arc"
#  [9] "arch"  "arm"   "art"   "ash"   "ass"   "axe"   "babe"  "back"
# [17] "badge" "bag"   "bail"  "bale"
```

Model input

```r
# Library
library(ndl)
# Generate cues
lexicon$Cues = orthoCoding(lexicon$Word, grams=2)
head(lexicon$Cues)
# [1] "#a_ac_ce_e#"       "#a_ag_ge_e#"       "#a_ai_id_de_e#"
# [4] "#a_ai_ir_r#"       "#a_ai_is_sl_le_e#" "#a_al_le_e#"
# Generate outcomes
lexicon$Outcomes = lexicon$Word
head(lexicon$Outcomes)
# [1] "ace"   "age"   "aide"  "air"   "aisle" "ale"
```

Training

```
# Estimate association strengths
weights = estimateWeights(lexicon)
round(weights[c("#a","ac","ce","e#"),1:5],4)
#          ace    age    aide     air   aisle
# #a  0.0094 0.2318  0.0058  0.1959  0.0024
# ac  0.0074 0.0094  0.0013  0.0275  0.0026
# ce  0.0014 0.0010  0.0002 -0.0048 -0.0087
# e# -0.0002 0.0041 -0.0006 -0.0039  0.0082
```

Associations

```
# View association strengths
rev(sort(weights["#q",]))[1:5]
#      queen      quest      guard      sense        set
# 0.3256534 0.2208410 0.1594041 0.1287231 0.1087365
rev(sort(weights["z#",]))[1:5]
#       blitz       waltz        tree         set         bin
# 0.34037104 0.14623702 0.10135350 0.07666841 0.07323060
```

Activations

- The activation (a_i) of a semantic outcome (O_i) given its input cues (C_k) is defined as:

$$a_i = \sum_{j \in \{C_k\}} V_{ji}$$

Activations

```
# Estimate activations
acts = estimateActivations(lexicon,weights)
rownames(acts$activationMatrix) = lexicon$Word
# View activations
rev(sort(acts$activationMatrix["view",]))[1:5]
#        view        vice        crew      friend       screw
# 0.970707258 0.026417128 0.020631840 0.013063277 0.008228428
rev(sort(acts$activationMatrix["vase",]))[1:5]
#        van       case       base       vase        set
# 0.91271410 0.55226661 0.13997458 0.06093920 0.04803283
rev(sort(acts$activationMatrix["yolk",]))[1:5]
#     youth      bulk      folk      silk      mode
# 0.9023739 0.3215203 0.3042634 0.2200161 0.1702520
```

Activations

```
# Define activation function
getActivation = function(word) {
  return(acts$activationMatrix[word,word])
}
# Extract activations
lexicon$Acts = as.numeric(sapply(lexicon$Word,getActivation))
head(lexicon[,c("Word","Acts")])
#     Word       Acts
# 1    ace 0.01804227
# 2    age 0.41010026
# 3   aide 0.03963205
# 4    air 0.48223668
# 5  aisle 0.08761536
# 6    ale 0.01855308
```

Reaction times

- Reaction times are inversely proportionally to a_i:

$$RT \propto \log(\frac{1}{a_i})$$

Reaction times

```
# Calculate simulated reaction times
lexicon$SimRT = log(1/lexicon$Acts)
# What is the correlation with observed reaction times?
cor(lexicon$RT,lexicon$SimRT)
# [1] 0.5237756
```

Simulations

- More frequency words are typically responded to faster
- Can we replicate this frequency effect?

Simulations

```
# Model for observed RTs
obs.lm = lm(RT ~ GoogleFrequency,data=lexicon)
# summary(obs.lm)
# ...
# Coefficients:
#                   Estimate Std. Error t value Pr(>|t|)
# (Intercept)       6.957556   0.017993  386.69   <2e-16 ***
# GoogleFrequency -0.034739   0.001197  -29.03   <2e-16 ***
# ---
# Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
# ...
cor(lexicon$RT,lexicon$GoogleFrequency)
# [1] -0.6285029
```

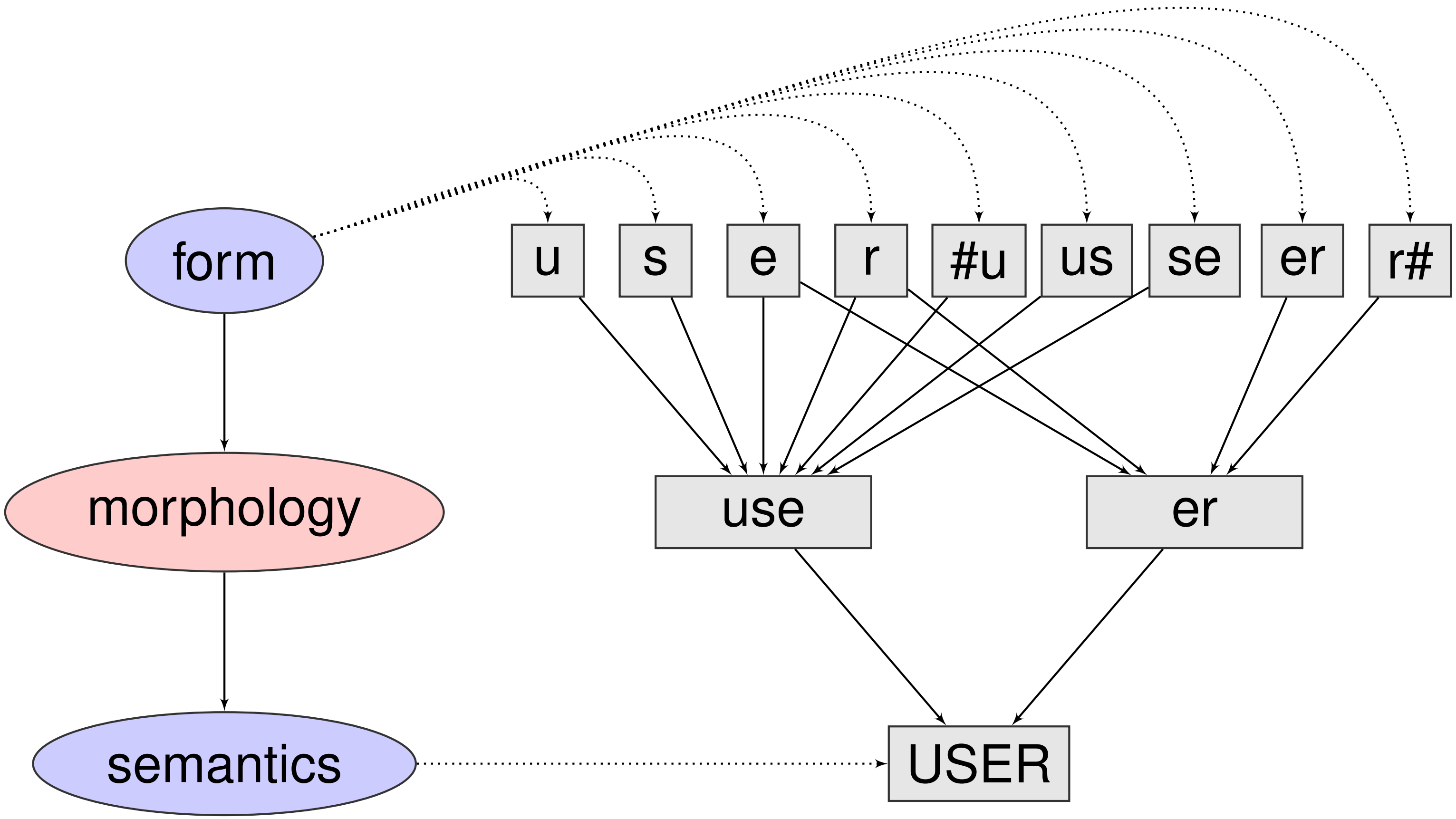
Simulations

```
# Model for simulated RTs
sim.lm = lm(SimRT ~ GoogleFrequency,data=lexicon)
# summary(sim.lm)
# ...
# Coefficients:
#                  Estimate Std. Error t value Pr(>|t|)
# (Intercept)       12.3172     0.2210   55.72   <2e-16 ***
# GoogleFrequency   -0.6662     0.0147  -45.32   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ...
cor(lexicon$SimRT,lexicon$GoogleFrequency)
# [1] -0.7836124
```

Applications

- Baayen et al. (2011)
- Hendrix, Ramscar & Baayen (2015)

Traditional morphological model



Why are morphemes problematic?

- Polysemy
- Example: $-s$:
  - *legs*
  - *walks*
  - *Harald's class*
- Example: $-er$:
  - *walker*
  - *greater*

Why are morphemes problematic?

- Gradient semantics:
  - Clear cases: $-$*ness* (e.g.; *greatness*)
  - Ambiguous cases: $-$*er* (e.g.; *walker*, *greater*)
  - Phonaesthemes: *gl*$-$ (e.g.; *glimmer*, *gleam*, ...)
  - “Partial support” (e.g.; $-$*er* in *archer*)

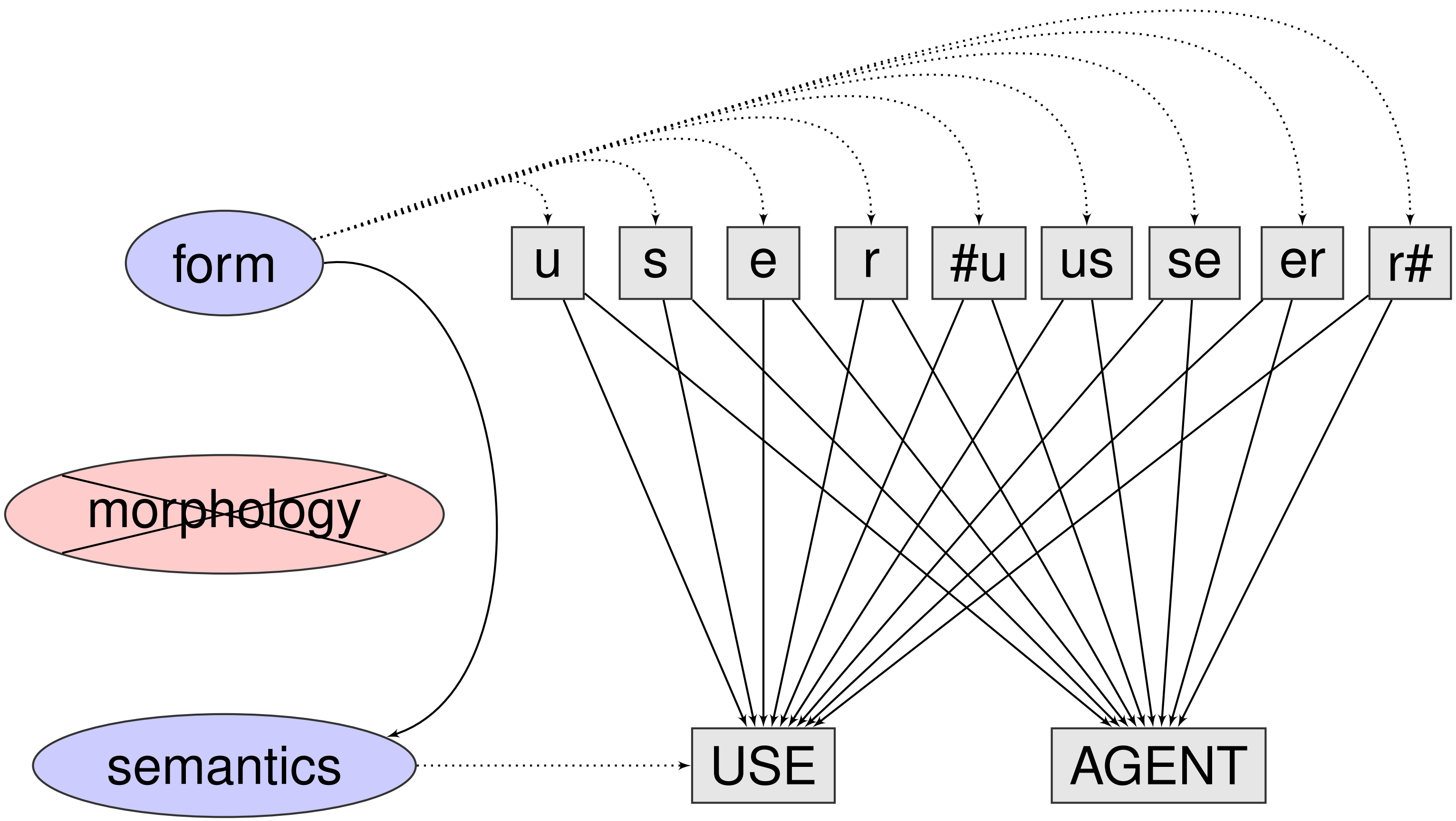
A-morphous morphology

- “the metaphor of morphology as a formal calculus with morphemes as basic symbols and morphological rules defining well-formed strings as well as providing a semantic interpretation, much as a pocket calculator interprets 2 + 3 as 5, is inappropriate”
- Conclusion: rage against the morpheme!

NDR

- Naive Discriminative Reader (Baayen et al. 2011)
- Two layers: orthography and semantics
- No morphological representations
- Fully compositional at the semantic level

NDR



NDR

- Training data:
  - 11,172,554 two and three-word phrases from the British National Corpus (BNC)
  - $26,441,155$ word tokens
  - $24,710$ word types
- Contextual training

Simulations

- Simple words
- Inflected words
- Derived words
- Pseudo-derived words
- Phrasal effects

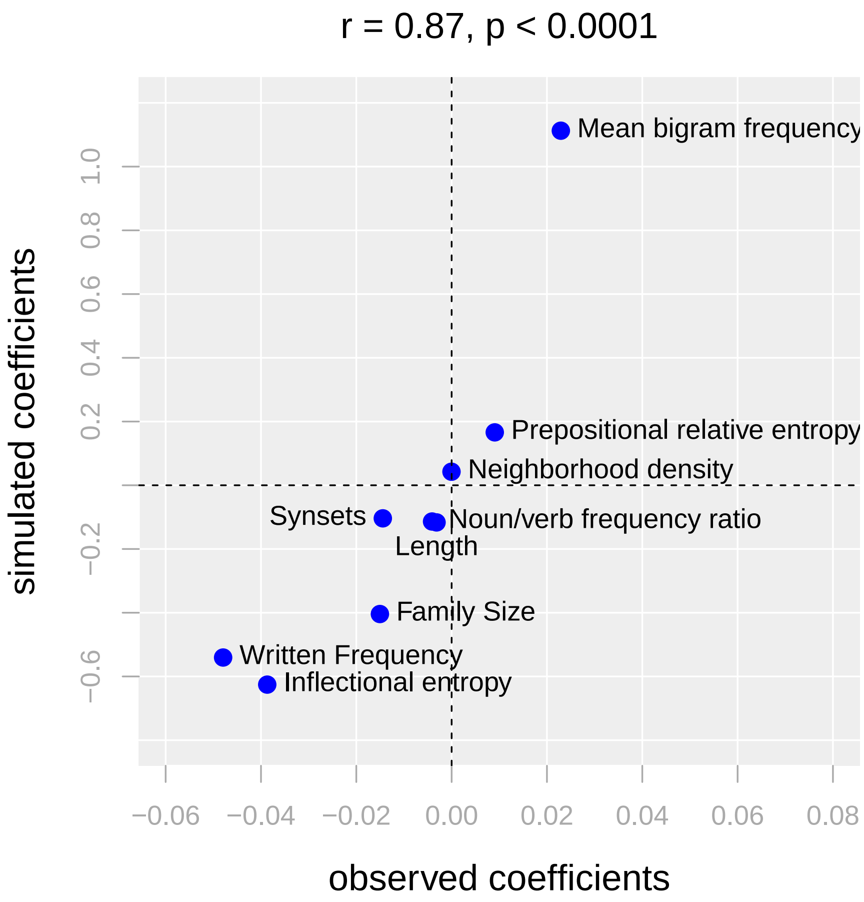
Simple words

- Stimuli: 1289 monomorphemic words (Baayen et al., 2006)
- Observed data: lexical decision latencies from the English Lexicon Project (ELP)
- Predictors:
  - Family Size
  - Written frequency, Noun/verb frequency ratio, Mean bigram frequency
  - Inflectional entropy
  - Length
  - Neighborhood density
  - Synonym count
  - Prepositional relative entropy

Simple words

- Correlation between observed and simulated RTs:
  $r = 0.56, p < 0.0001$
- Linear regression model on simulated and observed RTs

Simple words



Simple words

- The NDR successfully replicates a range of effects in single word reading
- Effect of morphological family size without any representations for morphologically complex words
- Effect of inflectional entropy without any representations for inflected words

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Simulations

- Simple words
- Inflected words
- Derived words
- Pseudo-derived words
- Phrasal effects

Inflected words

- Stimuli: present and past tense forms of 1,209 regular and 131 irregular monomorphemic verbs
- Observed data: lexical decision latencies from the English Lexicon Project (ELP)
- Predictors:
  - Regularity
  - Tense
  - Frequency
  - Length
  - Neighborhood density
  - Family size
  - Inflectional entropy

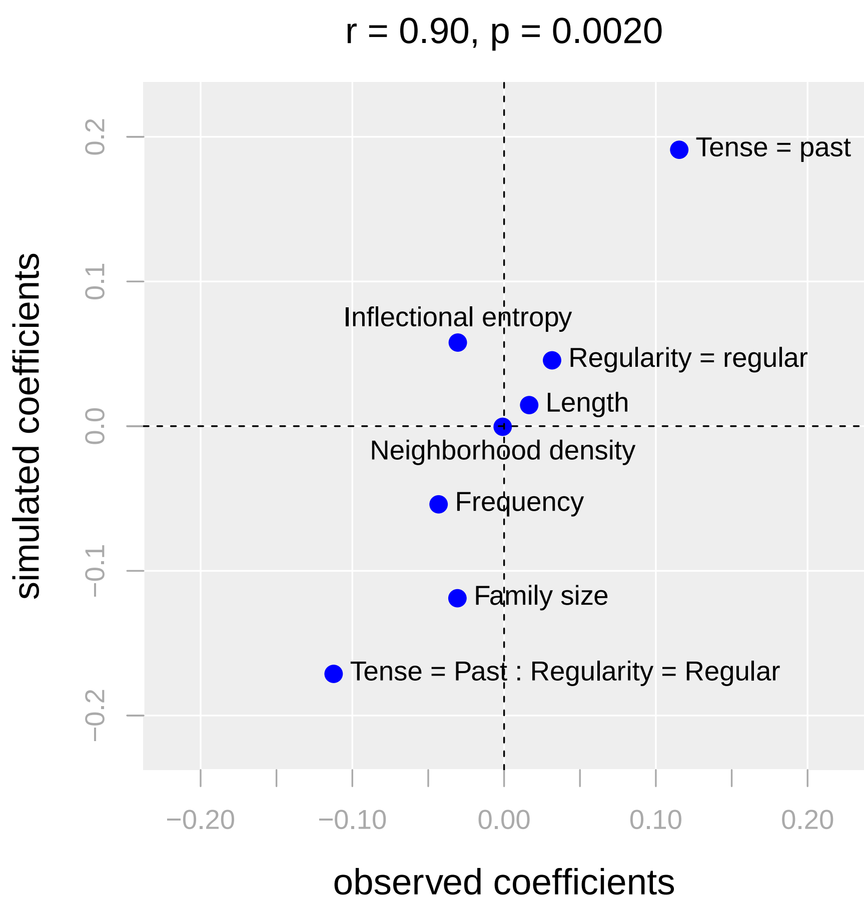
Inflected words

- Past tense represented as PAST
- Present tense not represented explicitly
- Simulated RTs are inversely proportional to a weighted combination of stem activation and tense activation

Inflected words

- Correlation between observed and simulated RTs: $r = 0.47, p < 0.0001$
- Mixed effects model

Inflected words

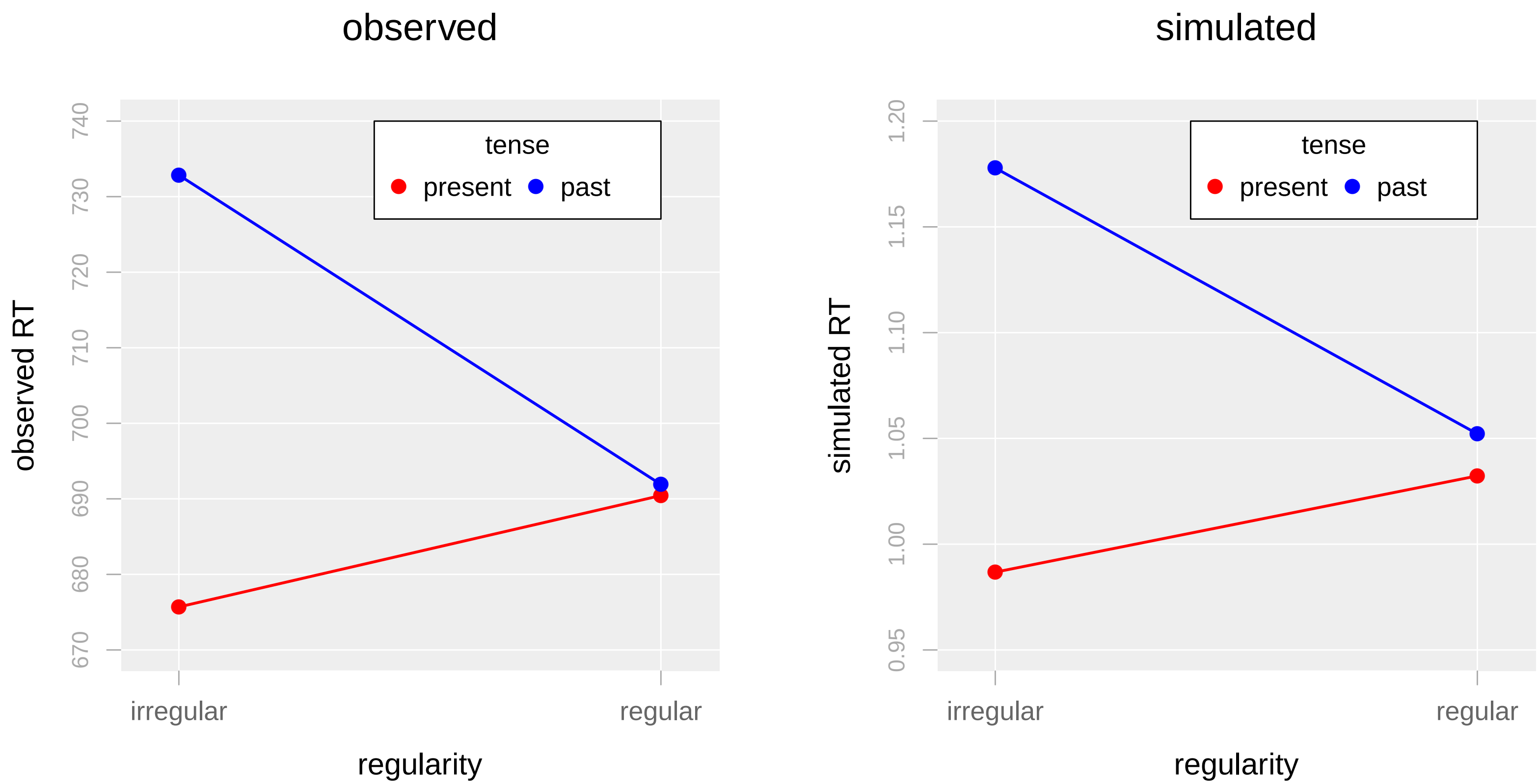


Inflected words

- NDR successfully captures effects of length, frequency, family size and regularity
- NDR incorrectly predicts facilitatory effect of inflectional entropy
- Training data did not provide information on aspectual meanings
- Empirical inflectional entropy does not match the learning experience of the model

UNIVERSITÄT TÜBINGEN

Inflected words



Inflected words

- Irregular past tenses have independent forms (e.g.; *bring* vs *brought*)
- Results for irregular verbs therefore reflect form frequencies
- Regular past tenses do not have independent forms (e.g. *walk* plus *PAST*)
- The suffix -ed has low cue validity as a past tense marker (*bed*, *red*, *greed*...)
- Results for regular verbs therefore reflect present tense frequencies

Simulations

- Simple words
- Inflected words
- Derived words
- Pseudo-derived words
- Phrasal effects

Derived words

- Lexical decision latencies
- Affix productivity

Derived words

- Derived words differ more substantially from base meanings than inflected words
- Example: *busy* versus *business*

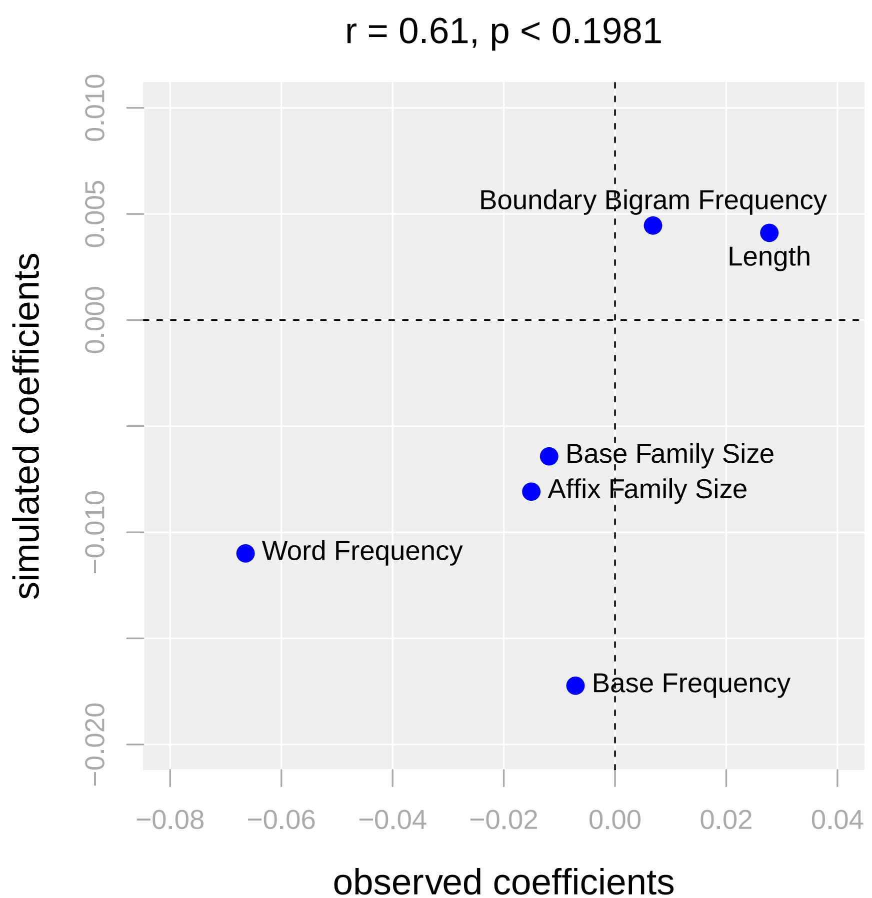
Derived words

- Stimuli: 3,003 derived words
- Observed data: lexical decision latencies from the ELP
- Predictors:
  - Frequency
  - Length
  - Frequency base, Family size base
  - Family size affix
  - Frequency boundary bigram
- Simulated RTs are inversely proportional to a weighted combination of stem activation and affix activation

Derived words

- Correlation between observed and simulated RTs: $r = 0.25, p < 0.0001$
- Mixed effects model

Derived words



Derived words

- Imbalance between whole-word frequency and base frequency; model is compositional even for opaque words (e.g.; *business*)
- Base frequency and base family size effects without any morphological parsing
- Whole-word frequency effect without any whole-word representations
- Boundary bigram frequency effect

Derived words

- Lexical decision latencies
- Affix productivity

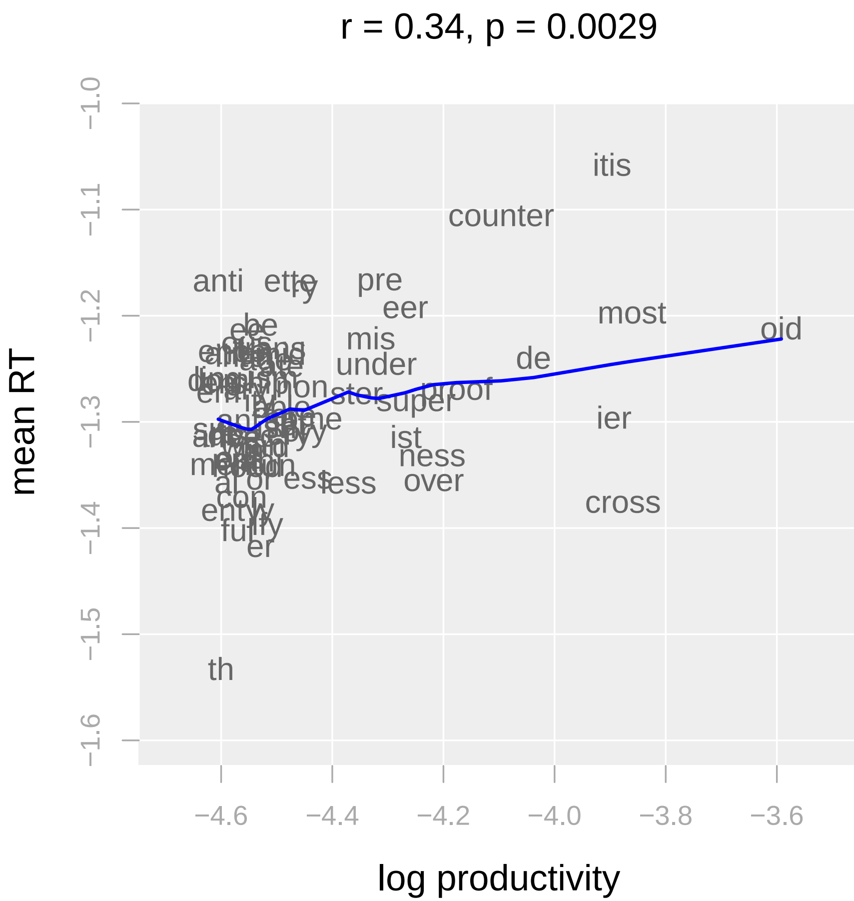
Derived words

- Affixes differ in their degree of productivity
- Example:
  - $-$*th*: 16 word types
  - $-$*ness*: 177 word types
- Measure of productivity:

  $$P = \frac{V_1}{N}$$

  where $V1$ is the number of types with token frequency 1 and N is the total number of tokens

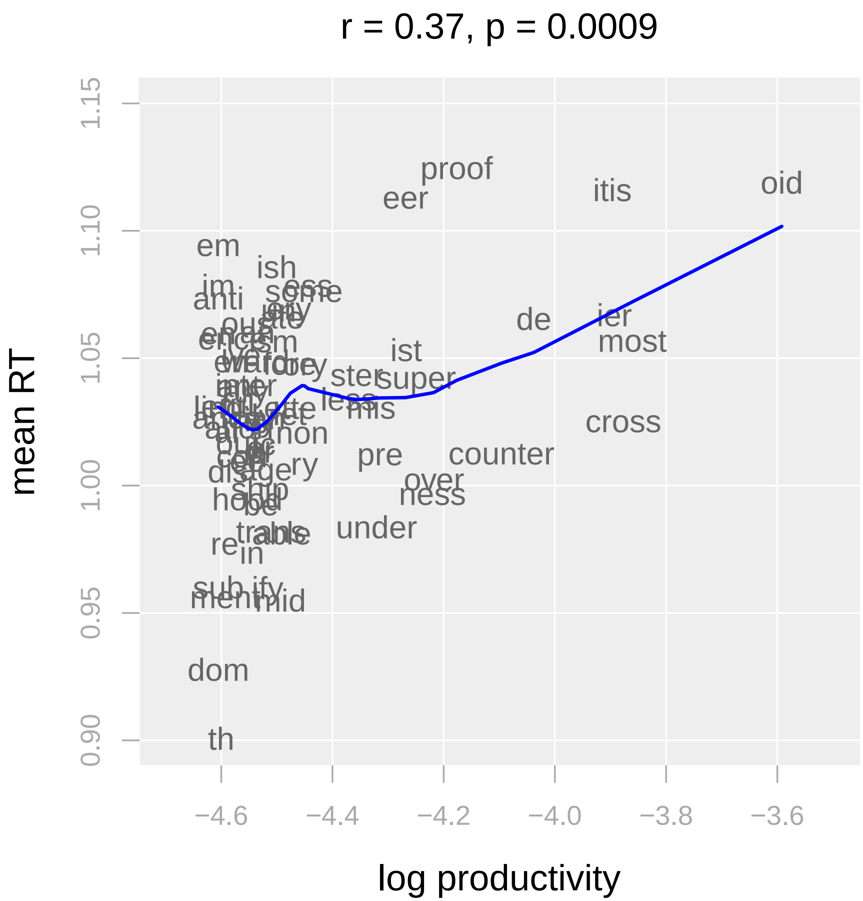
Derived words



Derived words

- More productive affixes lead to longer mean RTs
- Can the NDR replicate this effect?

Derived words



Derived words

- Why do less productive affixes correspond to shorter RTs?
- Occur in higher frequency words: better by-item learning
- Occur with fewer stems: better cue for these stems

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Simulations

- Simple words
- Inflected words
- Derived words
- Pseudo-derived words
- Phrasal effects

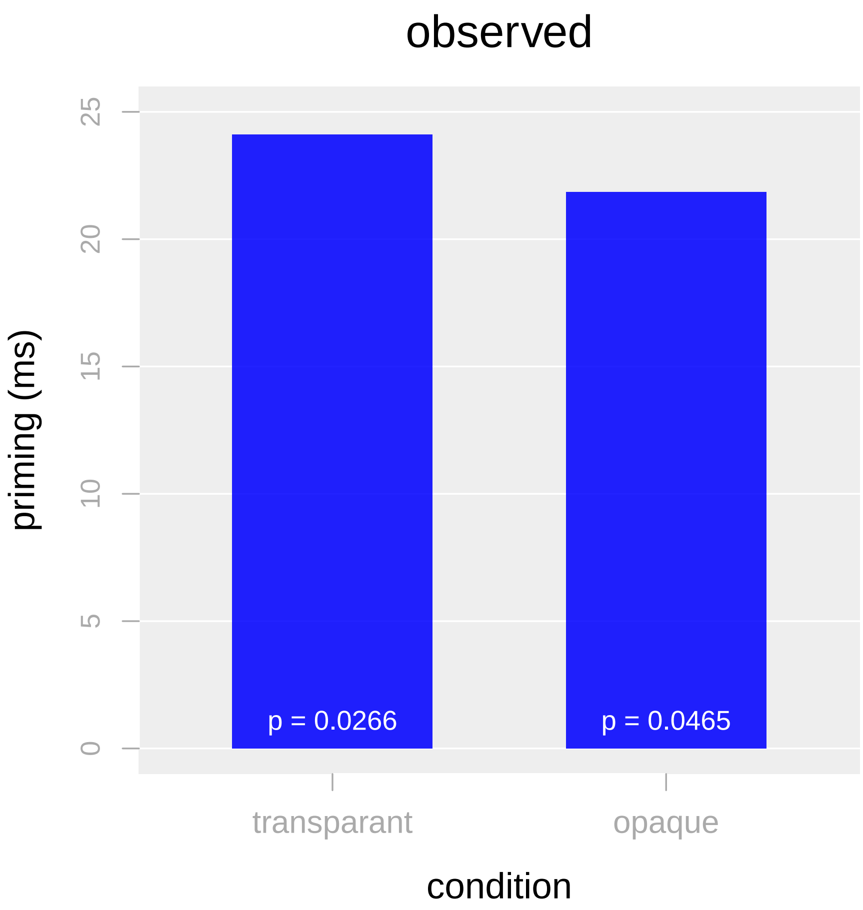
Pseudo-derived words

- Pseudo-derived words
- Phonaesthemes

Pseudo-derived words

- Stimuli: 294 prime-target pairs (Rastle et al., 2004)
- Three priming conditions:
  - Transparant: dealer-deal
  - Opaque: corner-corn
  - Orthographic: brothel-broth

Pseudo-derived words



Pseudo-derived words

- Interpretation: early form-based decomposition into morphemes
- Can the NDR model provide an alternative explanation?

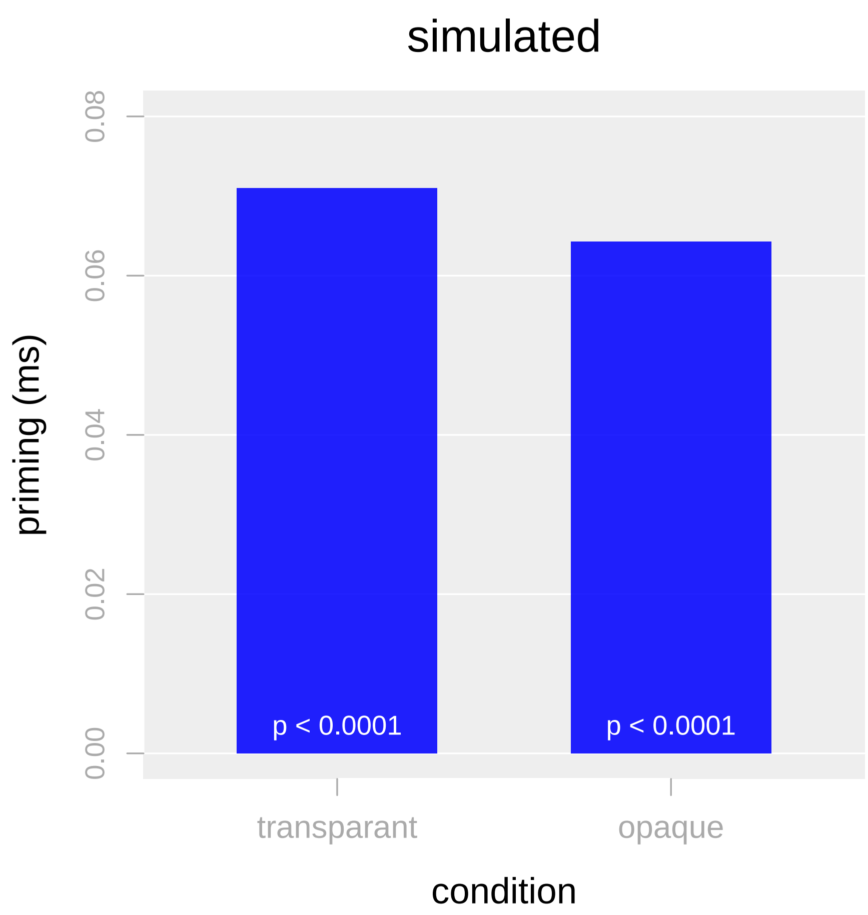
Pseudo-derived words

- Compound cue theory (Ratcliff & McKoon, 1988) to simulate priming effects
- Semantic decomposition when the suffix is synchronically active

Pseudo-derived words

Word	Type	Lexical Meaning	Suffix Meaning
archer	opaque	ARCHER	AGENT
fruitless	opaque	FRUITLESS	WITHOUT
trolley	opaque	TROLLEY	-
employer	transparent	EMPLOY	AGENT
cloudless	transparent	CLOUD	WITHOUT
brothel	form	BROTHEL	-
candidacy	form	CANDIDACY	-

Pseudo-derived words



Pseudo-derived words

- Morpho-orthographic effect without morpho-orthographic parsing
- For a majority of the opaque items, suffixes are semantically functional
- NDR model therefore learns associations between orthographic pseudo-suffixes and suffix meanings

Pseudo-derived words

- Pseudo-derived words
- Phonaesthemes

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Pseudo-derived words

- Phonaesthemes are frequent form-meaning mappings in the absence of a stem
- Example: *gl* in *glimmer*, *gloom*, *gleam*, *glow*, *glare*, *glint*
- Of all word tokens beginning with *gl*, 59.8% have meanings related to light or vision

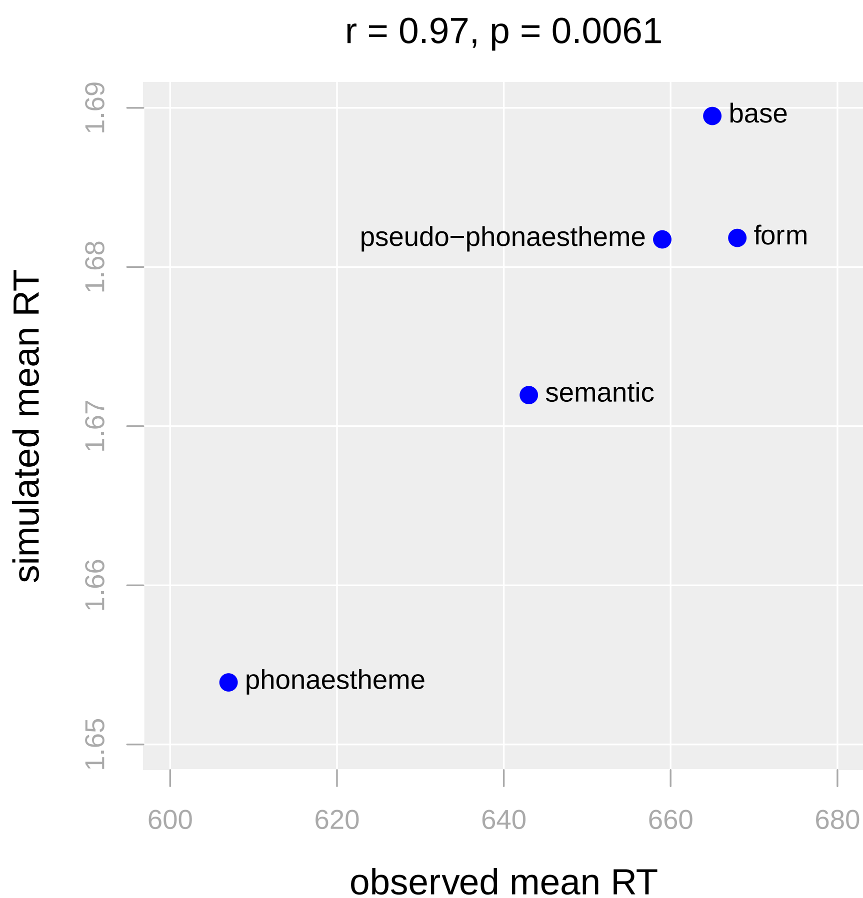
Pseudo-derived words

- Stimuli: 50 prime-target pairs (Bergen, 2004)
- Five priming conditions:
    - Phonaestheme: glimmer-gleam
    - Baseline: dial-ugly
    - Semantic: collar-button
    - Orthographic: druid-drip
    - Pseudo-phonaestheme: bleach-blank

Pseudo-derived words

- Compound cue theory (Ratcliff & McKoon, 1988) to simulate priming effects
- Phonaesthemes and pseudo-phonaesthemes encoded with two meanings:
  - *glimmer* $\rightarrow$ *GLIMMER*, *GL*
  - *gleam* $\rightarrow$ *GLEAM*, *GL*
- Semantically related words encoded with two meanings:
  - *collar* $\rightarrow$ *COLLAR*, *X*1
  - *button* $\rightarrow$ *BUTTON*, *X*1

Pseudo-derived words



EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Pseudo-derived words

- Morpheme-like effects can emerge without any morphemic representations

Simulations

- Simple words
- Inflected words
- Derived words
- Pseudo-derived words
- Phrasal effects

Phrasal effects

- Phrase frequency effects (Baayen, Hendrix & Ramscar, 2012)
- Phrasal paradigmatic effects

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Phrasal effects

- Arnon & Snider (2010): n-gram frequency effect over and above component frequency effects
- High frequency phrases are read faster than low frequency phrases:
    - “all over the place”
    - “all over the city”

Phrasal effects

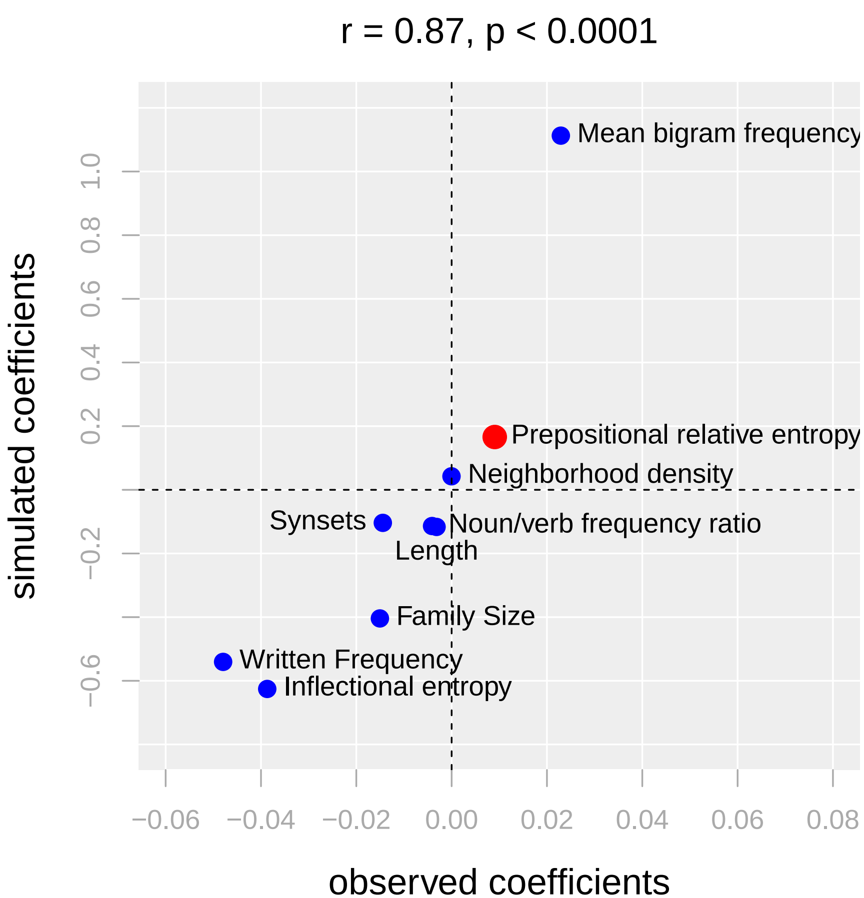
- NDR successfully simulates the phrase frequency effect
- Contextual learning
- The cues in *all*, *over* and *the* occur more frequently with *place* than with *city*

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Phrasal effects

- Phrase frequency effects (Baayen, Hendrix & Ramscar, 2012)
- Phrasal paradigmatic effects

Phrasal effects



Phrasal effects

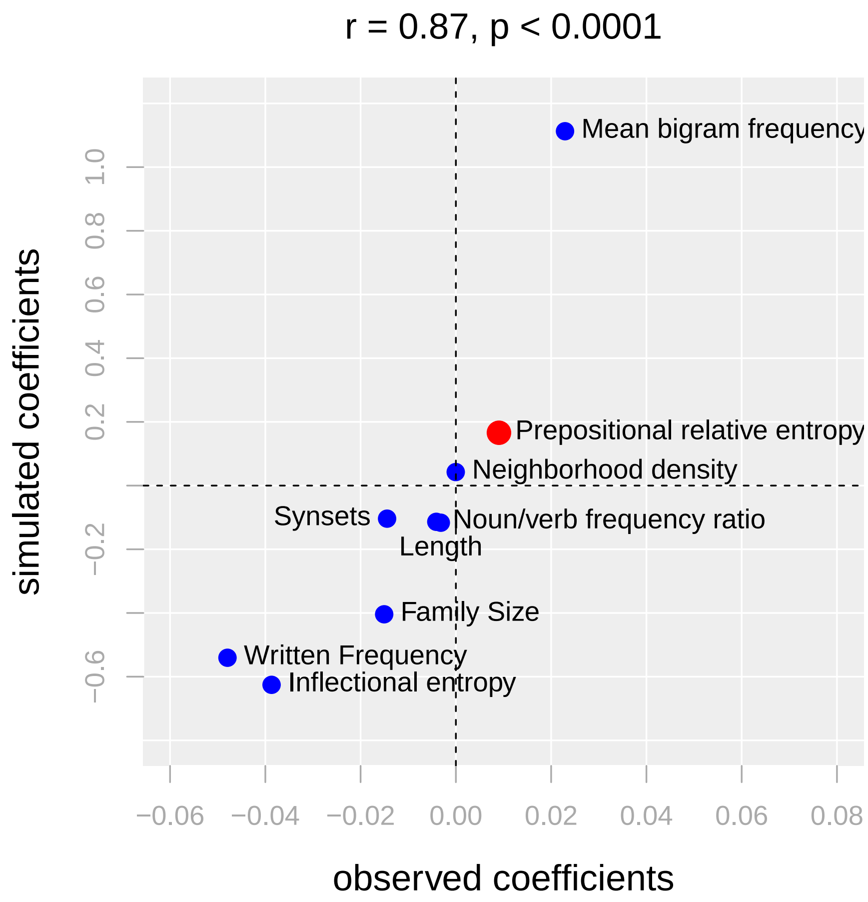
- Prepositional relative entropy is a measure of the prototypicality of a noun's use of prepositions
- It compares the frequency distribution of prepositions across all nouns with the frequency distribution of prepositions for a given noun
- The more similar the distributions the lower the prepositional relative entropy
- The more dissimilar the distributions the higher the prepositional relative entropy

Phrasal effects

Phrase	Freq	Prob	Preposition	Freq	Prob
on a plant	28608	0.279	*on*	177908042	0.372
in a plant	52579	0.513	*in*	253850053	0.531
under a plant	7346	0.072	*under*	10746880	0.022
above a plant	0	0.000	*above*	2517797	0.005
through a plant	0	0.000	*through*	3632886	0.008
behind a plant	760	0.007	*behind*	3979162	0.008
into a plant	13289	0.130	*into*	25279478	0.053

Relative entropy $= \sum_{i=1}^{n} p_i \log_2(p_i/q_i)$

Pseudo-derived words



Phrasal effects

- Prepositional relative entropy influences RTs in isolated lexical decision
- Higher relative entropy leads to longer RTs
- NDR successfully replicates this effect

Phrasal effects

- Phrase frequency and phrasal paradigmatic effects without any phrasal representations
- NDR successfully captures these effects through contextual learning
- Fits well with a gradient - rather than an absolute - distinction between morphology and syntax

Conclusions

- Morphological effects in the absence of morphological representations
- Consistent with a-morphous views on morphology (e.g.; Anderson, 1992; Blevins, 2003)

Conclusions

- The NDR is similar in spirit to the reading part of the triangle model (Seidenberg & Gonnermann, 2000)
- Both models map orthography onto semantics without any intermediate morphological representations
- Advantages NDR:
  - no back-propagation of error
  - no hidden layer units that can covertly represent morphology

Conclusions

- Computational engine is based on well-established discriminative learning algorithm
- Trained on realistic input data
- Parsimonious with respect to the number of representations
- Good fit to a wide range of experimental data

Applications

- Baayen et al. (2011)
- Hendrix, Ramscar & Baayen (2015)

Outline

- Introduction
- NDRa model
- Simulations
  - Overall model fit
  - Predictor simulations
  - Comparison to dual-route architecture
  - Pronunciation performance
- Conclusions

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Outline

- Introduction
- NDRa model
- Simulations
    - Overall model fit
    - Predictor simulations
    - Comparison to dual-route architecture
    - Pronunciation performance
- Conclusions

Introduction

- Existing models of reading aloud are dual-route models
- Lexical route
  - Orthography to phonology mapping is mediated by lexical representations
  - Responsible for reading known words (e.g.; *food*)
- Sub-lexical route
  - Direct orthography to phonology mapping
  - Responsible for reading unknown words (e.g.; *snood*)

Introduction

snood

Introduction

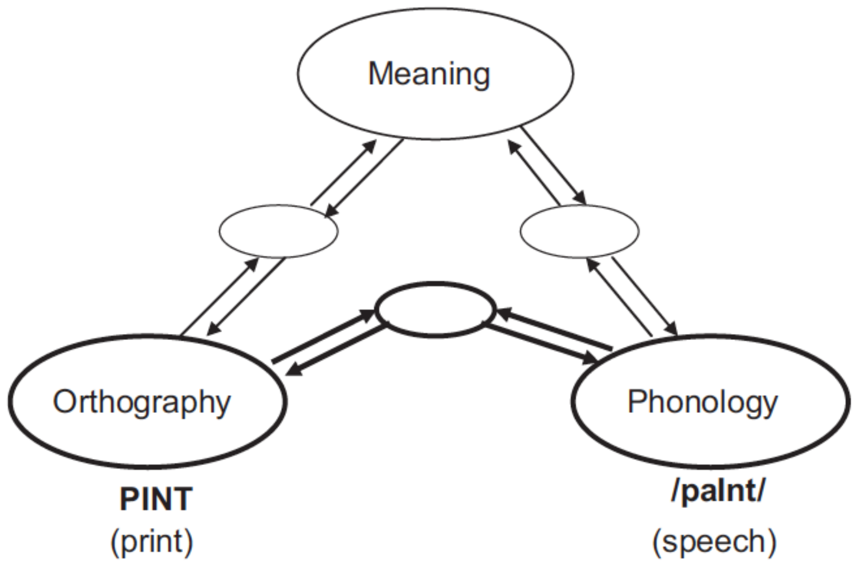
Introduction

- Examples of dual-route models:
  - Triangle model (Harm & Seidenberg, 2004)
  - DRC model (Coltheart et al., 2001)
  - CDP+ model (Perry et al., 2007)

Triangle model

- Connectionist model
- Three levels of description:
    - Orthography
    - Phonology
    - Semantics

Triangle model



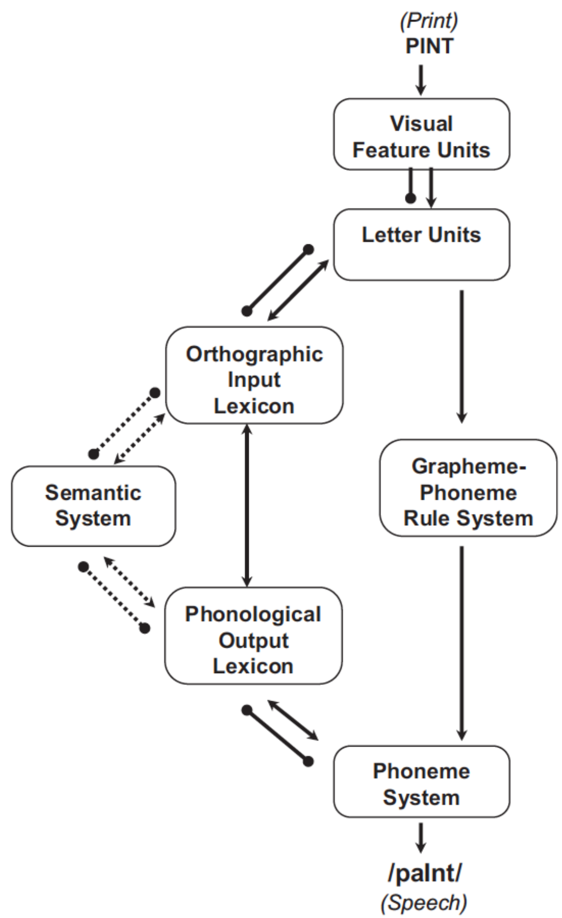
Triangle model

- What is represented by hidden layer units?
- Less explanatory power than CDP+ model

DRC model

- Lexical route: interactive activation model (McClelland & Rumelhart, 1981)
- Sub-lexical route: grapheme-to-phoneme conversion rules

DRC model



(Print)
PINT
Visual
Feature Units
Letter Units
Orthographic
Input
Lexicon
Semantic
System
Grapheme-
Phoneme
Rule System
Phonological
Output
Lexicon
Phoneme
System
/paInt/
(Speech)

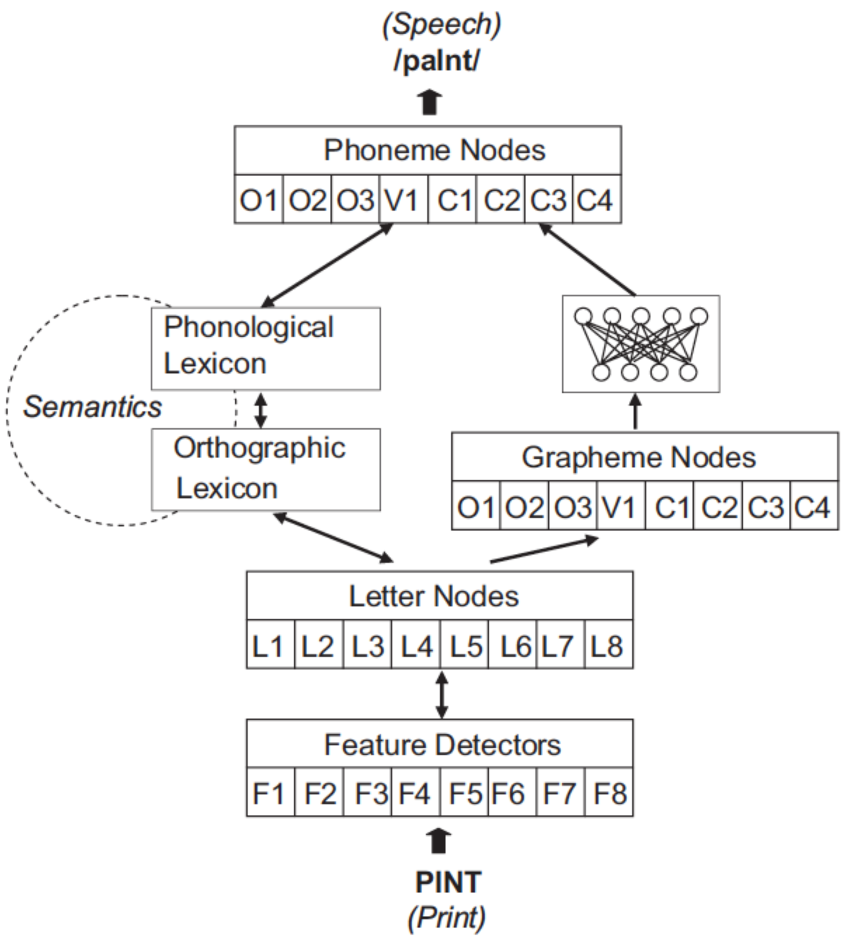
DRC model

- Ignores the problem of learning in both routes
- Poor performance compared to newer models of reading aloud

CDP+ model

- Successor of DRC model
- Hybrid model:
  - Lexical route: interactive activation model
  - Sub-lexical route: discriminative learning network (Zorzi et al., 1998)

CPD+ model



(Speech)
/paInt/
Phoneme Nodes
O1 O2 O3 V1 C1 C2 C3 C4
Phonological Lexicon
Semantics
Orthographic Lexicon
Grapheme Nodes
O1 O2 O3 V1 C1 C2 C3 C4
Letter Nodes
L1 L2 L3 L4 L5 L6 L7 L8
Feature Detectors
F1 F2 F3 F4 F5 F6 F7 F8
PINT
(Print)

CDP+ model

- Performs an order of magnitude better than other existing models of reading aloud
- Ignores the problem of learning in the lexical route

Introduction

- Can a learning network improve the performance of the lexical route?
- Is a sub-lexical route really necessary?

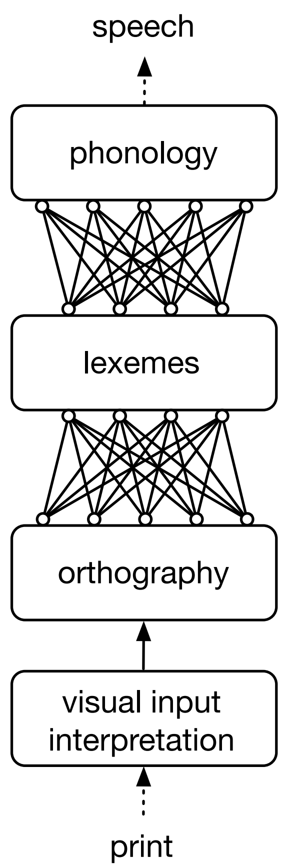
Outline

- Introduction
- NDRa model
- Simulations
  - Overall model fit
  - Predictor simulations
  - Comparison to dual-route architecture
  - Pronunciation performance
- Conclusions

NDRa

- Extension of the NDR model to reading aloud
- Single-route model: no sub-lexical route

NDRa model


speech
phonology
lexemes
orthography
visual input
interpretation
print

NDRa

- Visual input interpretation based on Manhattan city-block measure (Han & Kamber, 2000)
- More complex visual patterns should take longer to decode
- Complexity of a letter is inversely proportional to the similarity of that letter to all other letters
- Complexity of a word is the summed complexity of all component letters

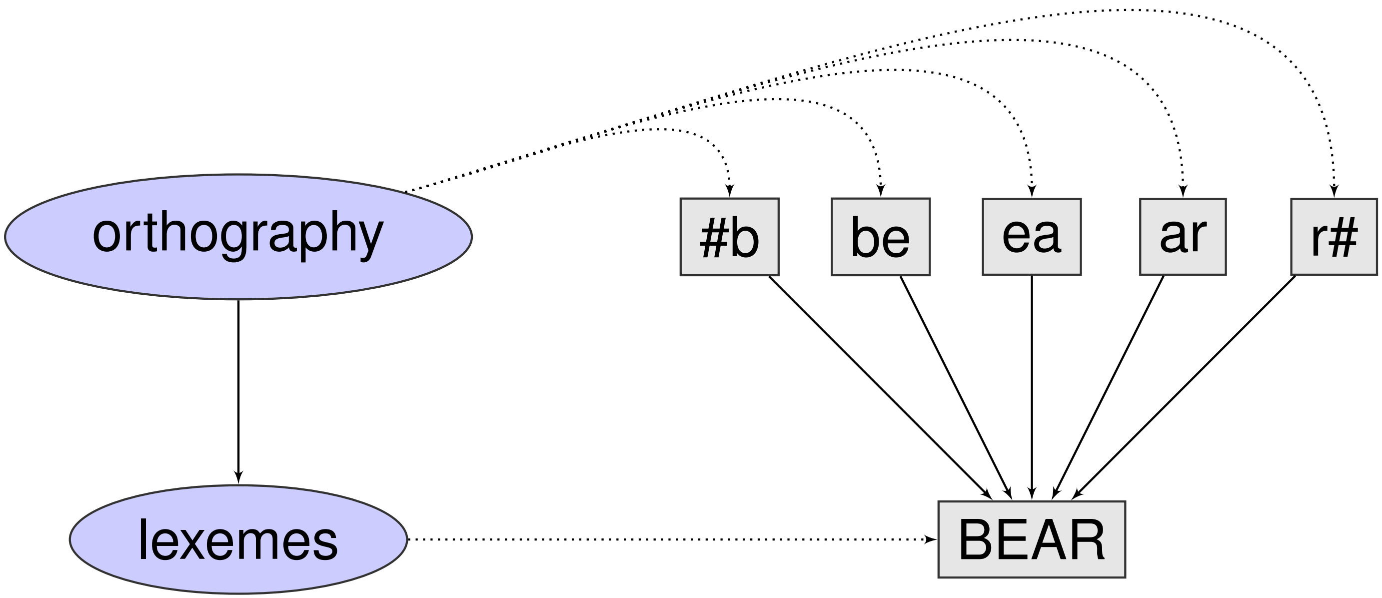
NDRa

- Two discriminative learning networks:
  - Orthography to lexemes
  - Lexemes to phonology

NDRa

- Orthography to lexeme network:
    - Original NDR model (Baayen et al., 2011)
    - Input units: letters and letter bigrams
      (e.g.; *#b, be, ea, ar, r#*)
    - Outcomes: lexemes (e.g.; *BEAR*)

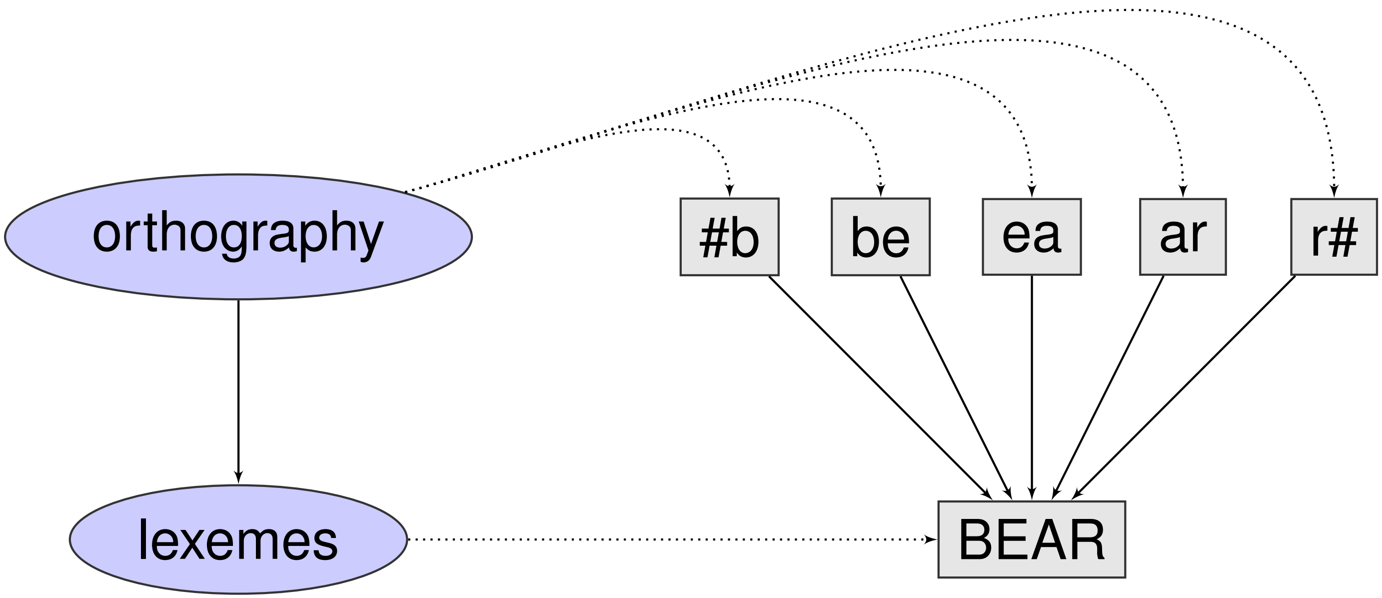
NDRa



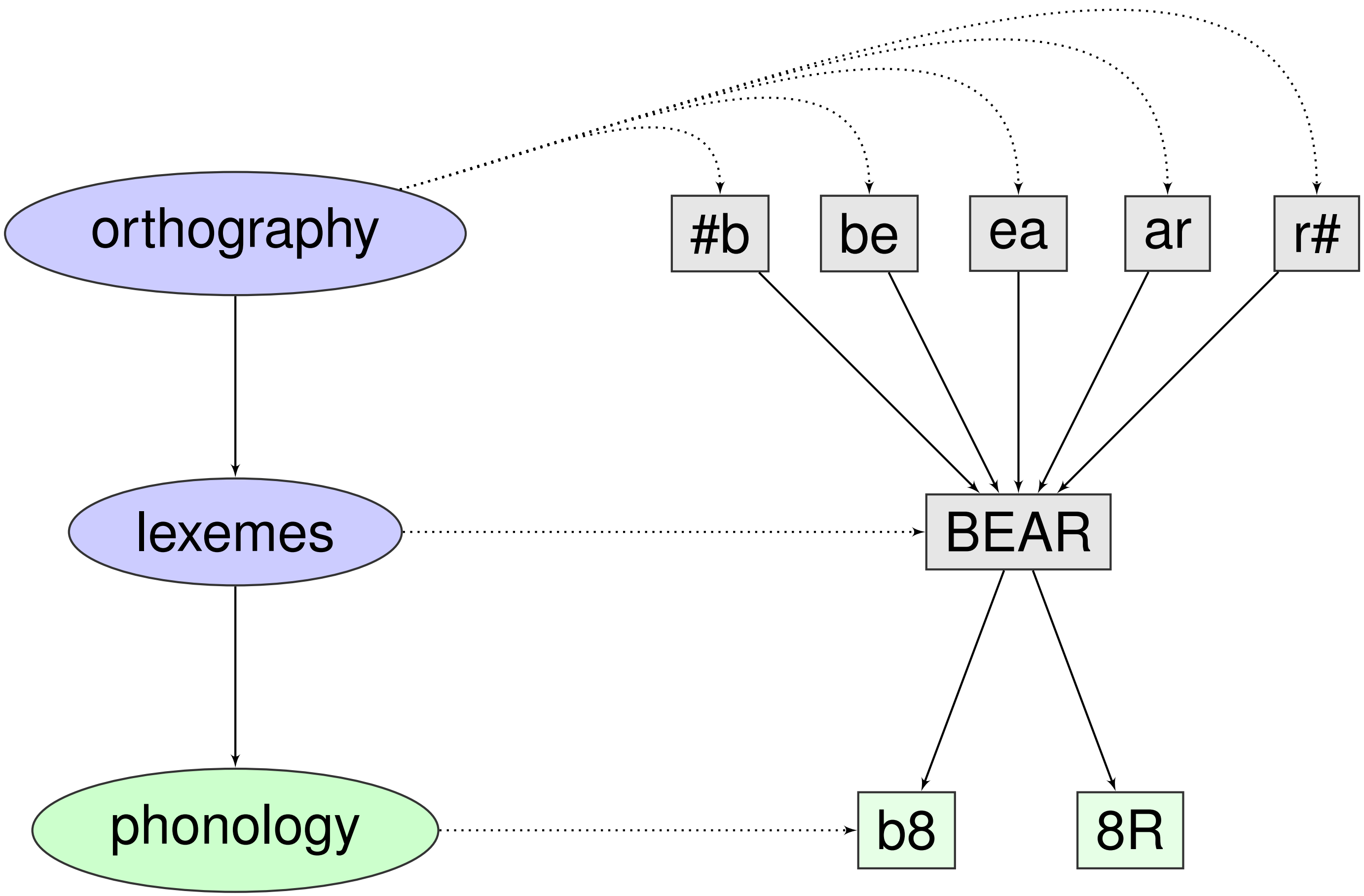
NDRa

- Lexeme to phonology network:
  - New in the NDRa model
  - Input units: lexemes (e.g.; *BEAR*)
  - Outcomes: demi-syllables (e.g.; *b8, 8r*)

NDRa



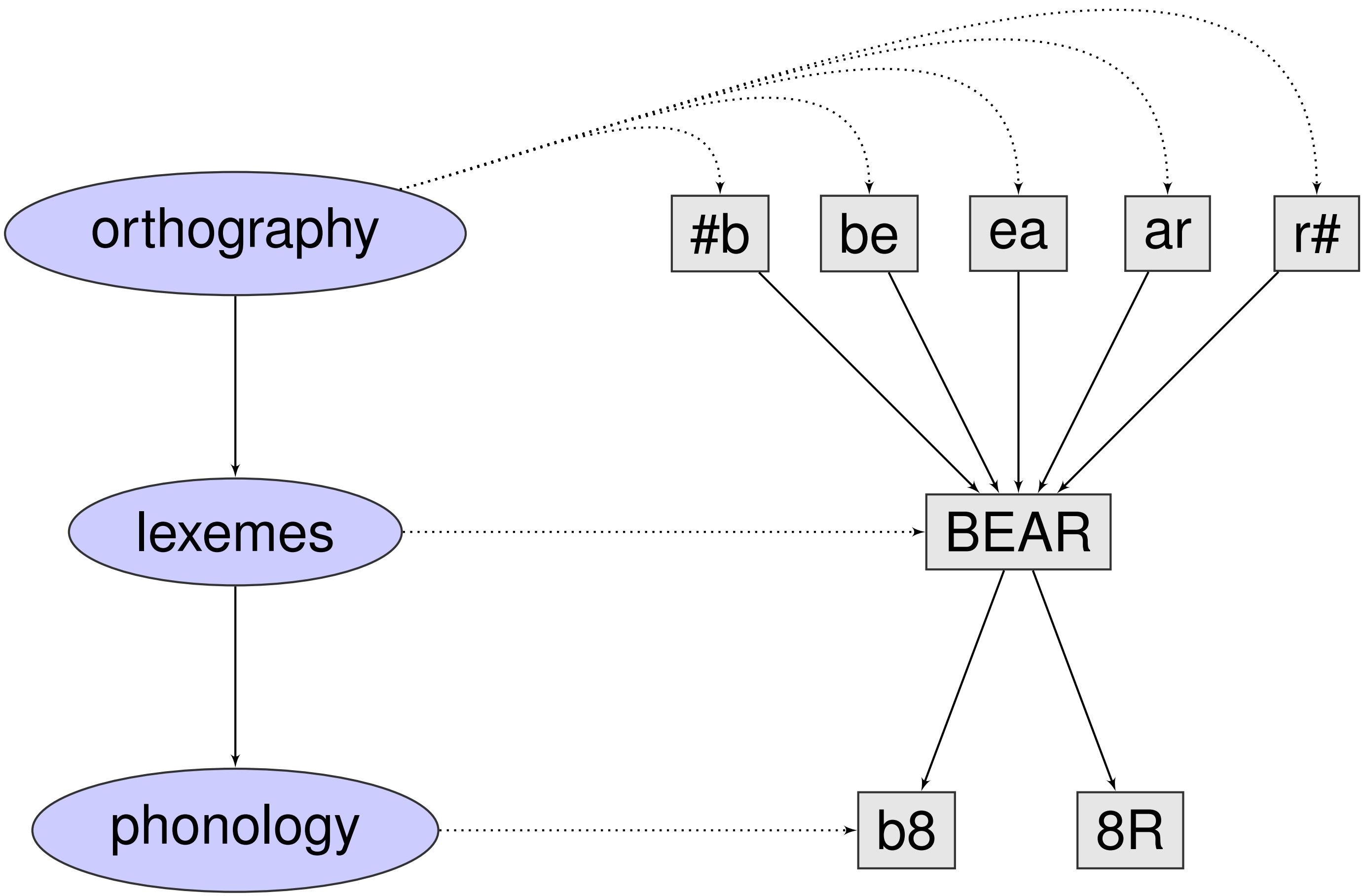
NDRa



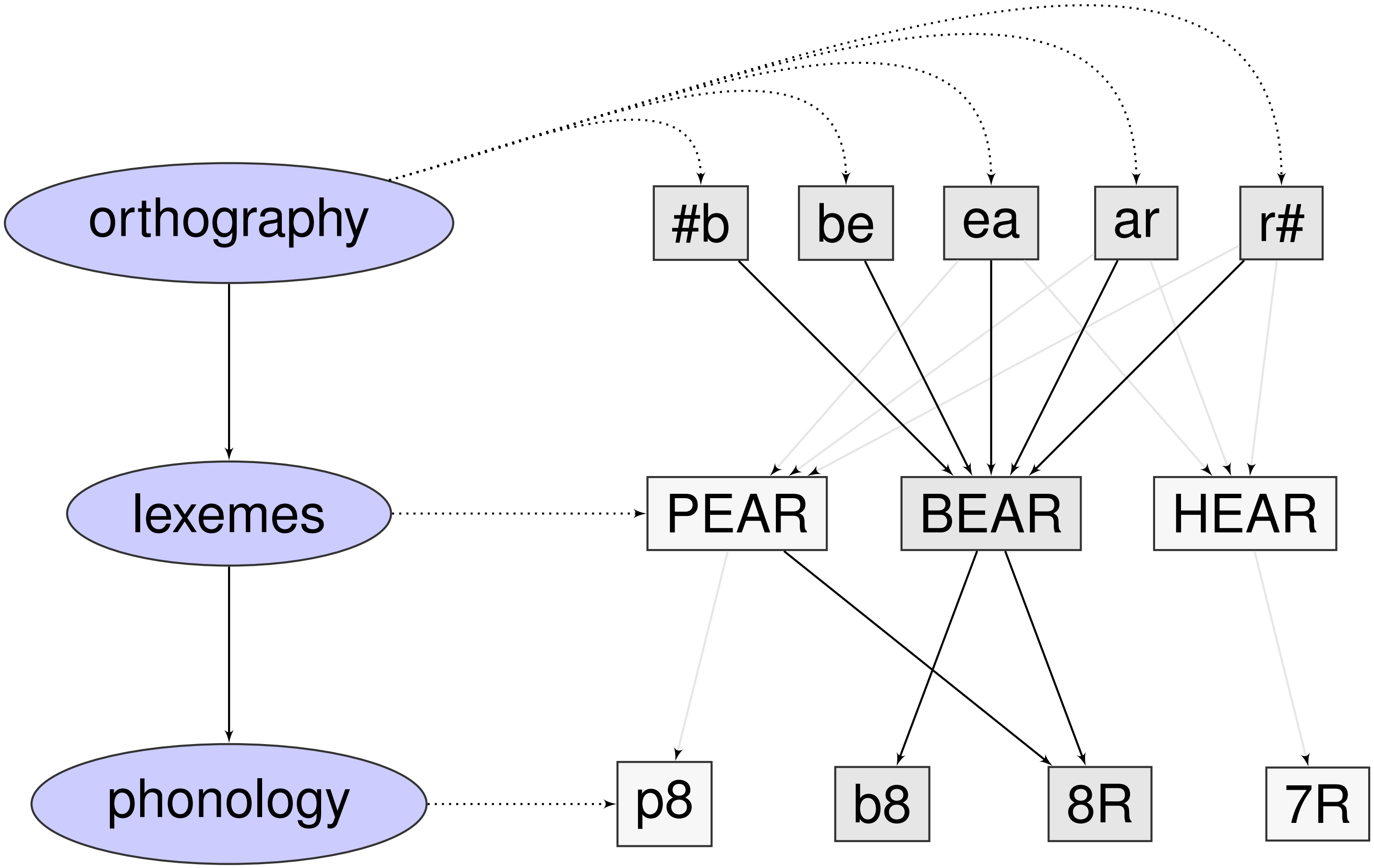
NDRa

- Orthographic units activate not only the lexeme of the target word, but also the lexemes of orthographic neighbors of the target word
- Phonological units are activated by the lexeme of the target word as well as by the lexemes of the activated orthographic neighbors

NDRa



NDRa



orthography
lexemes
phonology
#b
be
ea
ar
r#
PEAR
BEAR
HEAR
p8
b8
8R
7R

NDRa

- Given the activation a_t from the target lexemes and the activations $a_{1,\dots,n}$ from the lexemes of co-activated orthographic neighbors, the total activation of a demi-syllable k is defined as:

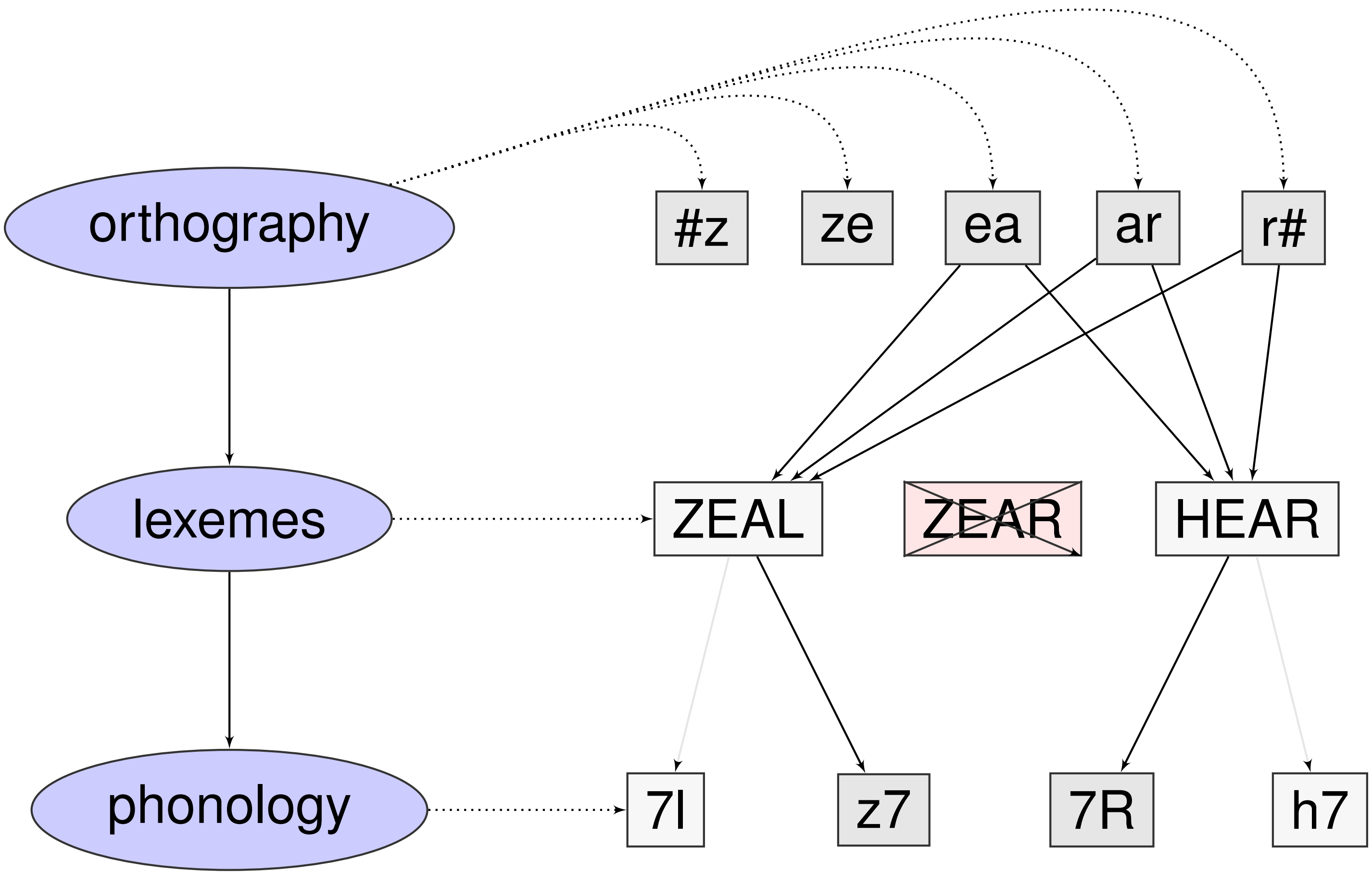
  $$ActPhon_k = w_{lex} * a_t + \sum_{i=1}^{n} w_i * a_i$$

  where w_i is the amount of activation that lexical neighbor lexeme i received from the orthography of the target word and w_{lex} is the relative weight of the activation from the target word lexeme as compared to the activation from the lexical neighbor lexemes

NDRa

- Unknown words and non-words are processed by the same architecture
- No lexico-semantic representations exist for non-words
- Pronunciation is therefore mediated only by the activations of orthographic neighbors

NDRa



orthography
#z
ze
ea
ar
r#
lexemes
ZEAL
ZEAR
HEAR
phonology
7l
z7
7R
h7

NDRa

- NDRa models reading aloud of monosyllabic words
- Choice problem: demi-syllables have to be combined in the right order
- Modeled through the entropy over the activations of the first and second demi-syllable

NDRa

- Simulated reaction times are proportional to a weighted multiplicative integration of:
  - the complexity of the visual input
  - the activation of the target word lexeme
  - the activation of the demi-syllables of the target word
  - the entropy over the demi-syllable activations
- $RT \propto \frac{\text{Complexity}^{w1}}{\text{ActLexeme}^{w_2} * \text{ActPhon}_1^{w_3} * \text{ActPhon}_2^{w_4} * \text{H}^{w_5}}$

  where $w_{1,\dots,5}$ are weight parameters that determine the relative contribution of each source of information

Outline

- Introduction
- NDRa model
- Simulations
  - Overall model fit
  - Predictor simulations
  - Comparison to dual-route architecture
  - Pronunciation performance
- Conclusions

Simulations

- 2524 mono-syllabic words
- 1822 non-words
  - 912 regular non-words
  - 910 pseudohomophones (e.g.; "bloo")
- 16 linguistic predictors

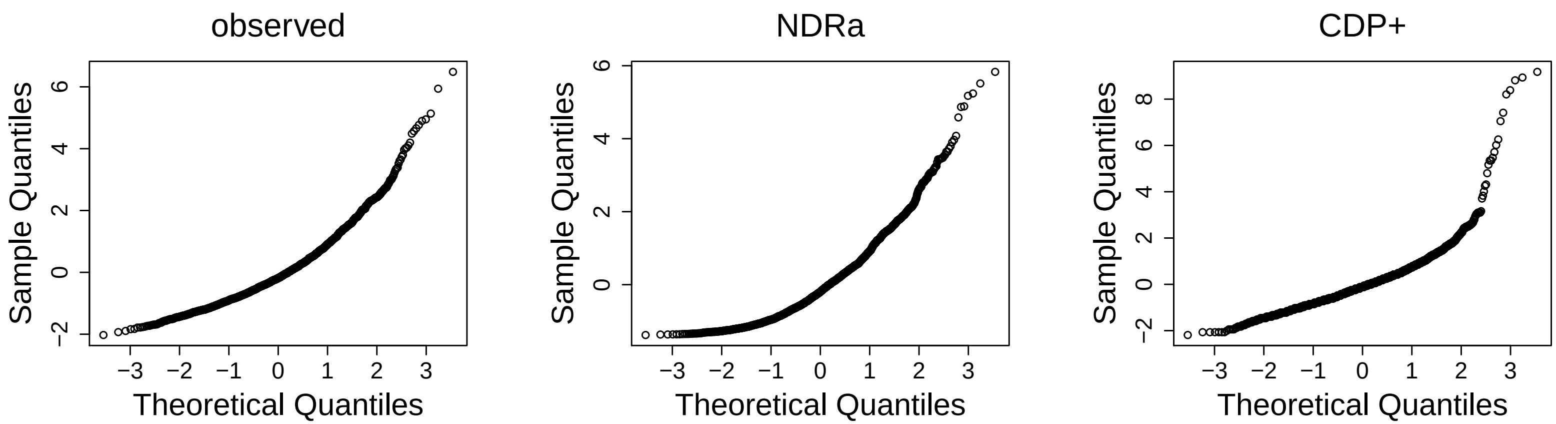
Outline

- Introduction
- NDRa model
- Simulations
    - Overall model fit
    - Predictor simulations
    - Comparison to dual-route architecture
    - Pronunciation performance
- Conclusions

Overall model fit

- Comparison of simulated latencies and observed ELP naming latencies:
  - $r = 0.50$ for NDRa, 0.49 for CDP+
  - AIC much better for NDRa
  - Latency distribution much better for NDRa

Overall model fit



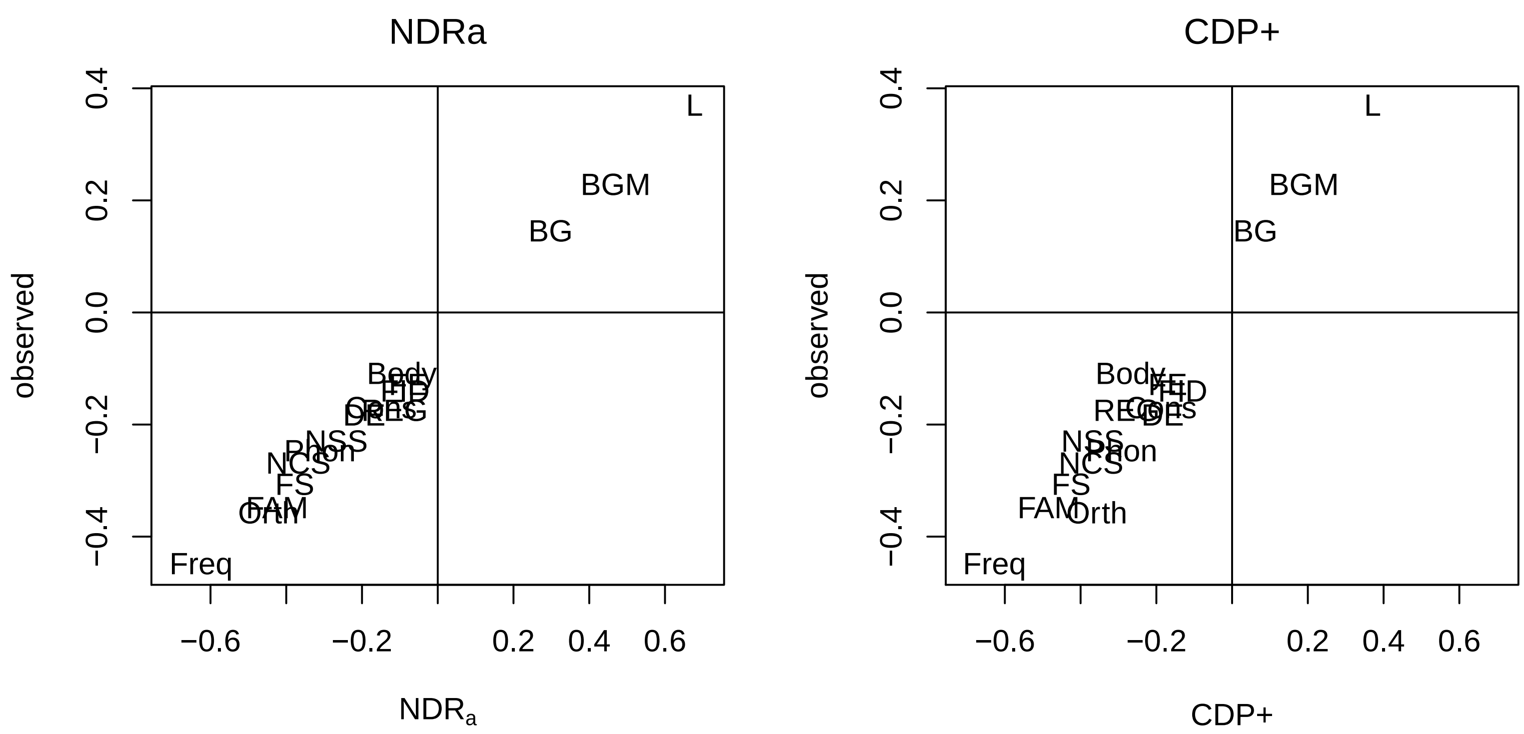
Outline

- Introduction
- NDRa model
- Simulations
    - Overall model fit
    - Predictor simulations
    - Comparison to dual-route architecture
    - Pronunciation performance
- Conclusions

Predictor simulations

- How well do both models capture the effects of the 16 linguistic predictors?
- Fit a separate linear model for each predictor
- Compare β coefficients between models for normalized observed and simulated latencies

Predictor simulations



L	=	length	FE	=	friends-enemies ratio	DE	=	derivational entropy	FS	=	family size
BGM	=	mean bigram freq.	FID	=	freq. initial diphone	NSS	=	number of simplex synsets	FAM	=	familiarity
BG	=	summed bigram freq.	Cons	=	consistency	Phon	=	phon. neighb. size	Orth	=	orth. neighb. size
Body	=	body neighb. size	REG	=	regularity	NCS	=	number of complex synsets	Freq	=	frequency

Predictor simulations

- Excellent performance for both models
- Nearly perfect correlation with observed β coefficients for the NDRa model ($r = 0.997$)
- CDP+ model seems to have problems with the relative contribution of neighborhood measures

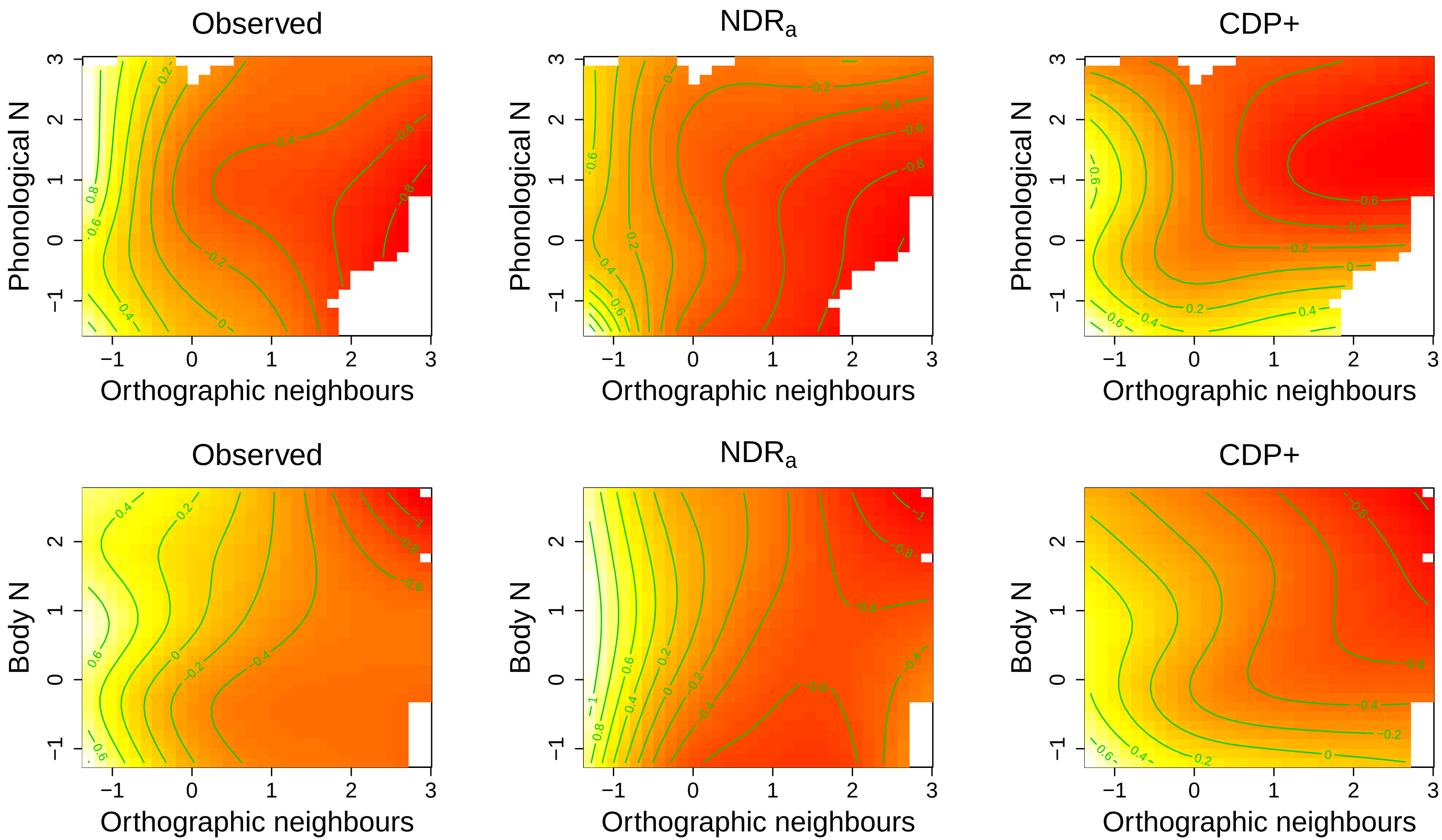
Predictor simulations: neighborhood measures

- Effects of three neighborhood measures have been documented:
  - Orthographic neighborhood (e.g.; *bear* - *pear*, *bear* - *hear*, *bear* - *bead*)
  - Phonological neighborhood (e.g.; *bear* - *pear*, *bear* - *hair*, *bear* - *bail*)
  - Body neighborhood (e.g.; *bear* - *pear*, *bear* - *wear*)

Predictor simulations: neighborhood measures

- Both models successfully capture the non-linear effects of all three predictors in isolation fairly well
- What about the non-linear interplay of the neighborhood measures?
- Find out using tensor product GAMs on simulated and observed latencies

Predictor simulations

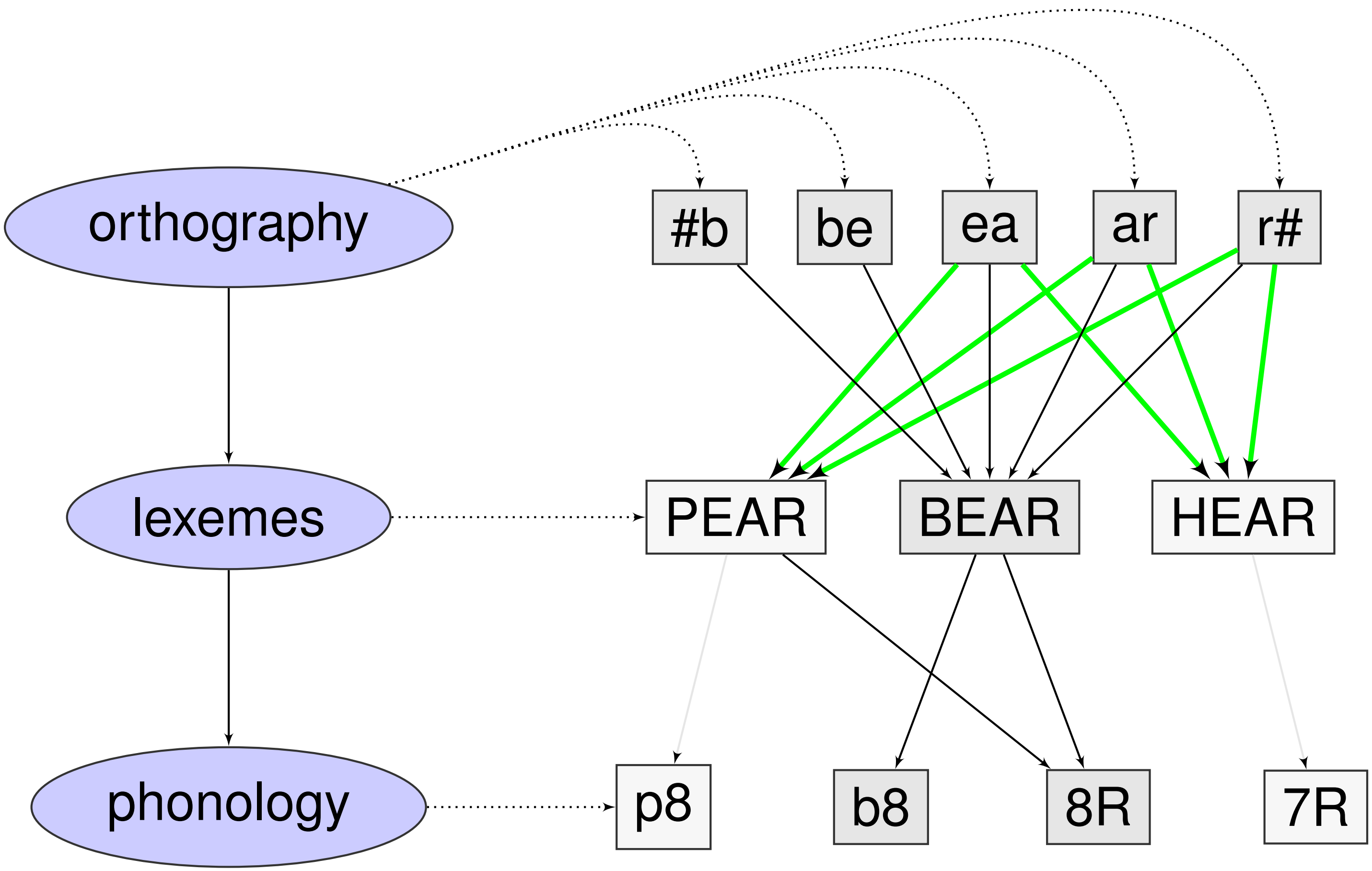


Predictor simulations: neighborhood measures

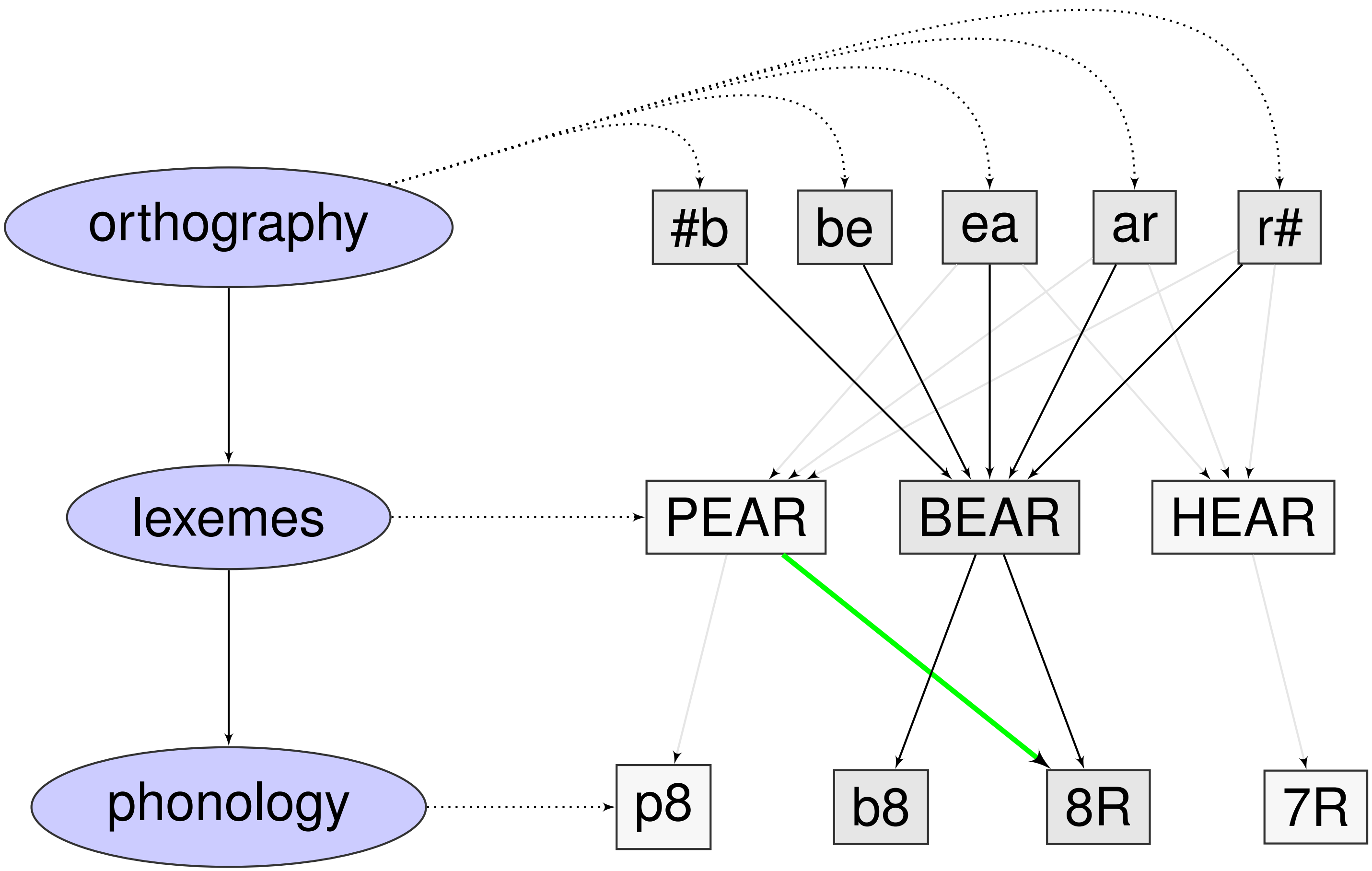
- Neighborhood effects are primarily orthographic neighborhood effects
- The NDR model correctly predicts the non-linear interplay of the neighborhood measures

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

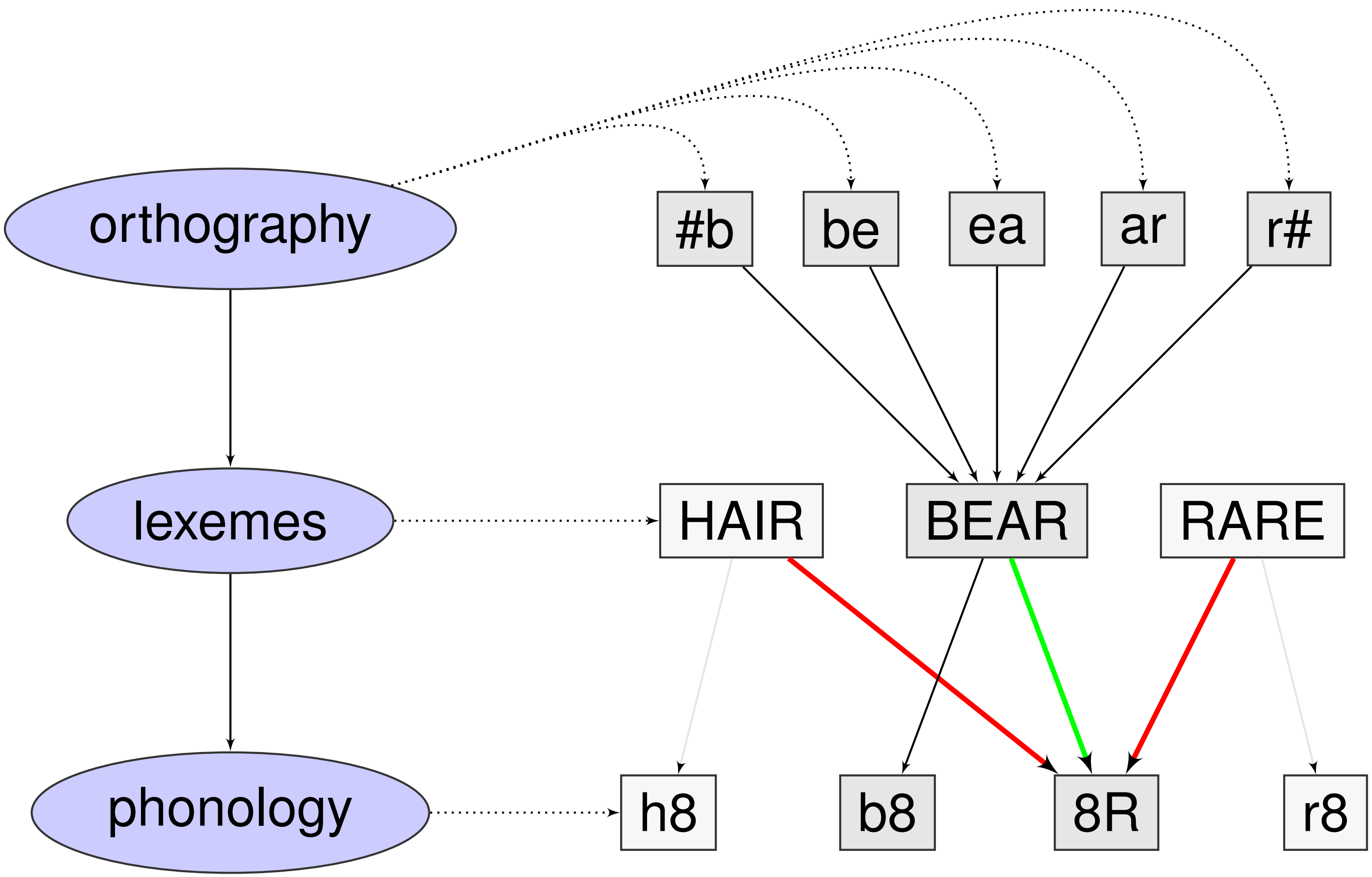
Predictor simulations: neighborhood measures



Predictor simulations: neighborhood measures



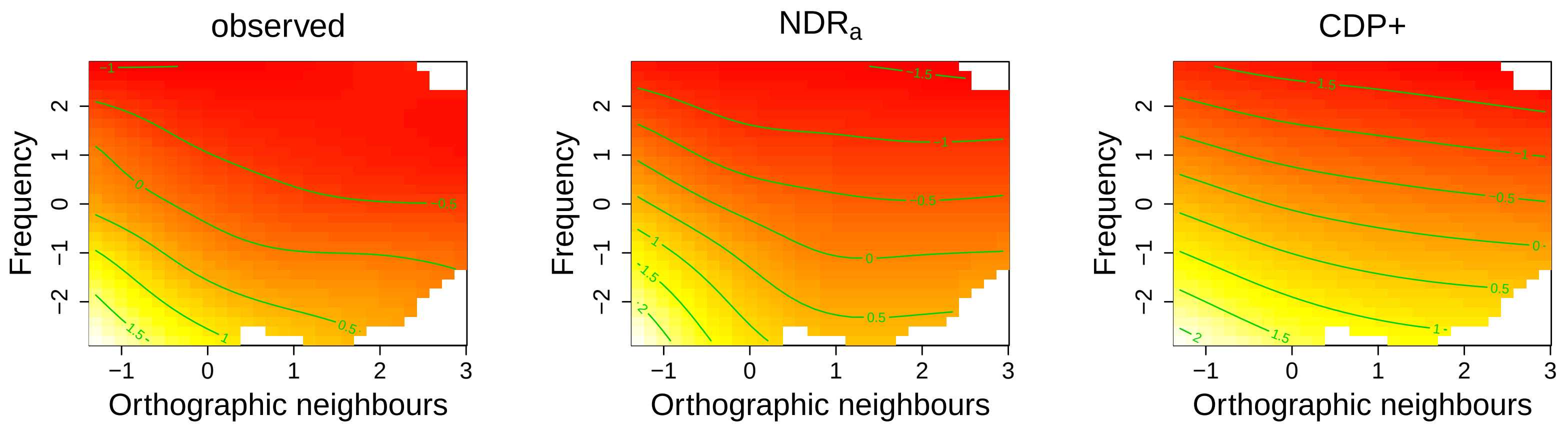
Predictor simulations: neighborhood measures



Predictor simulations: neighborhood measures

- Orthographic neighborhood density interacts with frequency in observed naming latencies
- Only low-frequency words show a neighborhood density effect
- Can the NDRa capture this interaction?

Predictor simulations: neighborhood measures



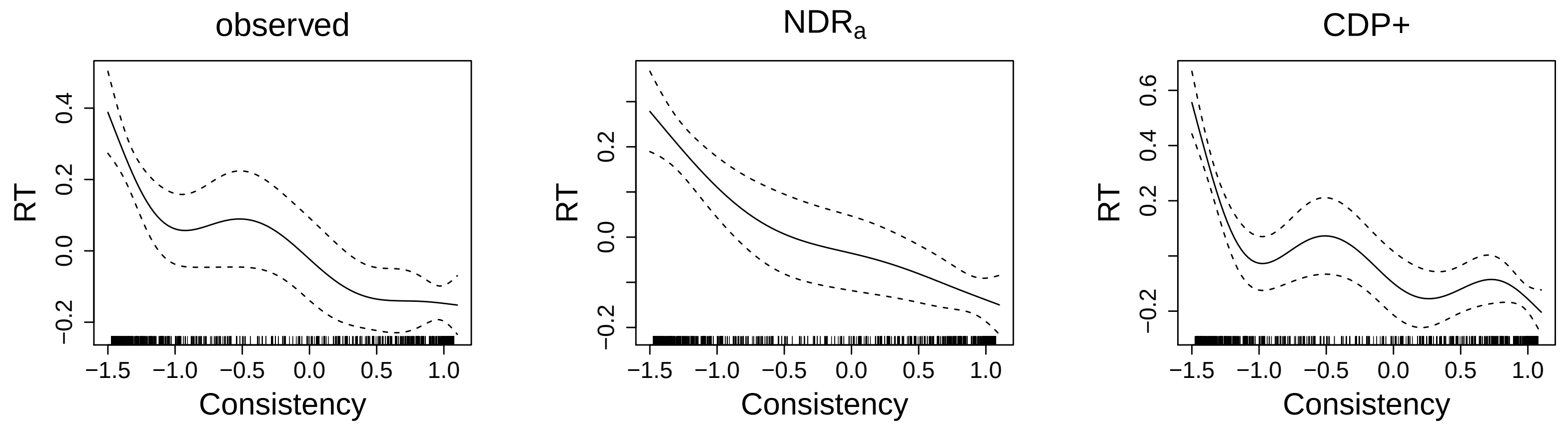
Predictor simulations: consistency

- The orthography to phonology mapping can be consistent or inconsistent
- Consistent with *pear*: *bear*, *wear*
- Inconsistent with *pear*: *dear*, *fear*, *gear*, *hear*, *lear*, *near*, *rear*, *year*, ...
- Higher proportions of consistent word tokens correspond to shorter naming latencies

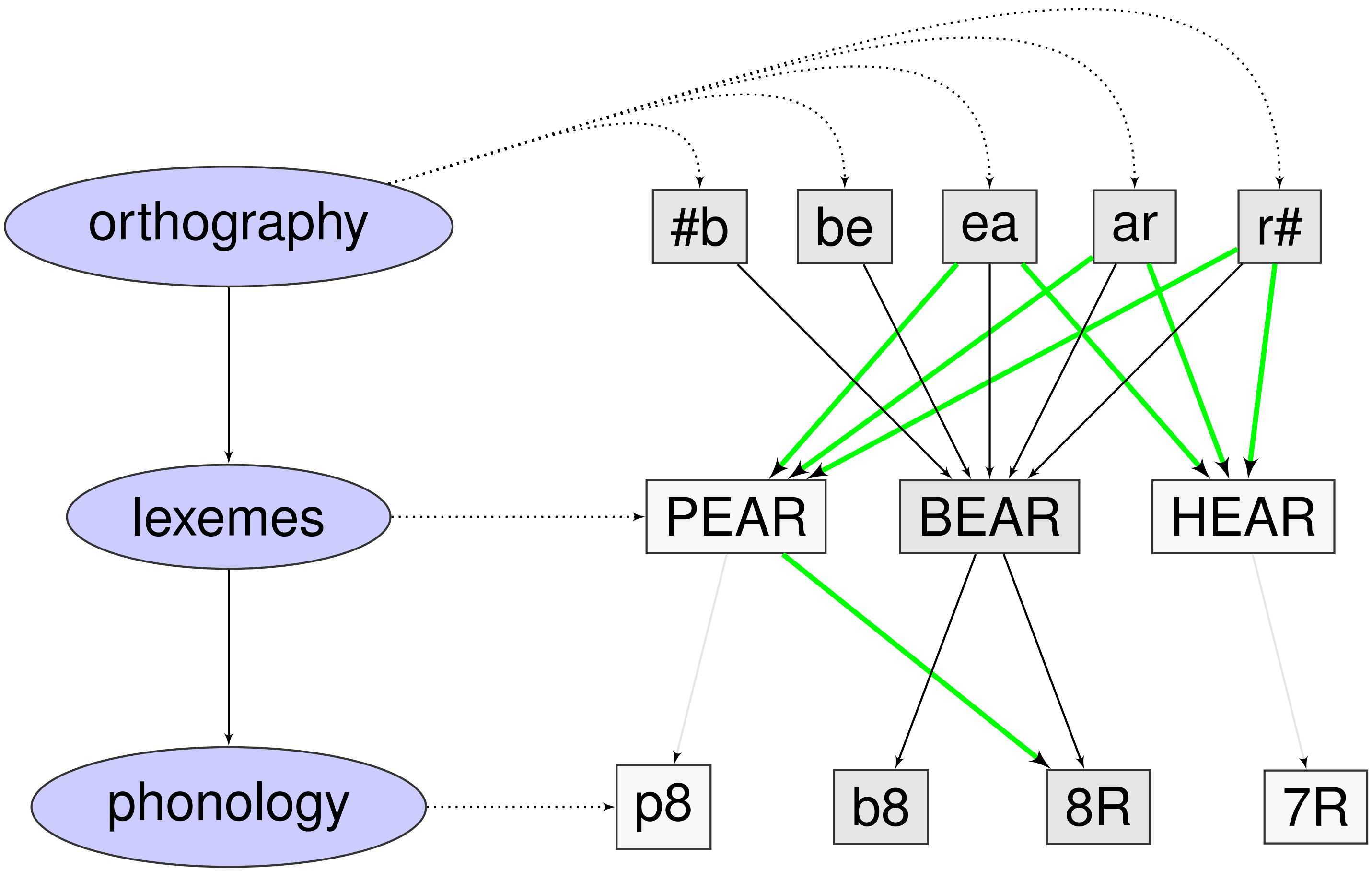
Predictor simulations: consistency

- Capturing consistency effects was a major advancement of the CPD+ model over the original DRC model
- In the CDP+ model consistency effects arise in the learning network in the sub-lexical route
- Can the single-route NDRa capture the effect of consistency?

Predictor simulations: consistency



Predictor simulations: consistency



Predictor simulations: non-words

- The NDRa successfully replicates a large number of effects in word naming
- How about non-word naming?

Predictor simulations: non-words

- Non-words naming effects captured by the NDRa include:
  - Non-words are read slower than real words
  - Non-word naming latencies increase linearly with length
  - Non-words with more orthographic neighbors are read faster
  - A higher proportion of consistent word tokens leads to shorter non-word naming latencies
  - Pseudohomophones are read faster than regular nonwords

Predictor simulations: non-words

- [Ramscar]Does the frequency of non-words help predict naming latencies?[/Ramscar]
- Re-analysis of naming latencies for non-words in McCann & Besner (1987)

Predictor simulations: non-words

- [Ramscar]Does the frequency of non-words help predict naming latencies?[/Ramscar]
- Re-analysis of naming latencies for non-words in McCann & Besner (1987)
- Frequency is the strongest predictor for non-word naming latencies

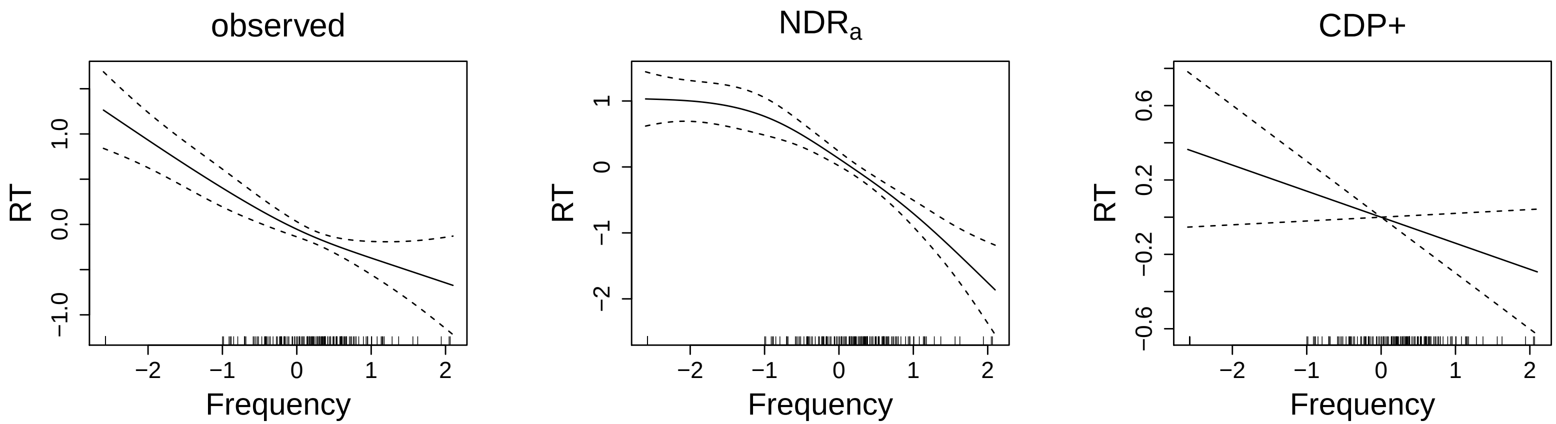
Predictor simulations: non-words

- Difference between words and non-words is graded rather than absolute
- Both words and non-words may or may not have a lexical representation in the mental lexicon of an individual language user
- The probability of a lexical representation is a function of the frequency of a word or non-word
- Fits well with the single-route architecture of the NDRa

Predictor simulations: non-words

- Simulation: retrain NDRa with Google frequencies for non-words
- Does the NDRa capture the effect of non-word frequency?

Predictor simulations: non-words



Outline

- Introduction
- NDRa model
- Simulations
  - Overall model fit
  - Predictor simulations
  - Comparison to dual-route architecture
  - Pronunciation performance
- Conclusions

Comparison to dual-route architecture

- The single-route NDRa model explains a wide range of experimental effects in both word and non-word naming
- Would a sub-lexical route further improve the performance of the NDRa?

Comparison to dual-route architecture

- Add a sub-lexical route to the NDRa
- Discriminative learning network from orthography to phonology
- Does this network help explain additional variance in the observed data?

Comparison to dual-route architecture

	NDR_a	NDR_a^2
Lexical route		
ActLexeme	5.011	3.231
$ActPhon_1$	5.989	6.003
$ActPhon_2$	12.259	11.499
H	7.520	7.077
Complexity	18.019	16.851
Non-lexical route		
$ActPhonSub_1$	NA	0.398
$ActPhonSub_1$	NA	1.114
HSub	NA	1.246

Comparison to dual-route architecture

- Components of the sub-lexical route do not help explain additional variance
- Correlation with observed naming latencies remains the same
- Conclusion: the addition of a sub-lexical route does not improve the performance of the NDRa model

Outline

- Introduction
- NDRa model
- Simulations
  - Overall model fit
  - Predictor simulations
  - Comparison to dual-route architecture
  - Pronunciation performance
- Conclusions

Pronunciation performance

- Naming latencies reflect bottom-up processes
- Discrimination learning captures bottom-up processing
- Response selection involves top-down processes
- The pre-frontal cortex plays an important role in response conflict resolution
- Functional architecture?

Pronunciation performance

- Provisionary checking mechanism
- Filter set of lexemes that activate demi-syllables based on orthographic overlap with the target word
- Real words: consider activation from target word lexeme only
- Non-words:
  - Initial demi-syllable: consider activation from words that share orthographic onset only
  - Second demi-syllable: consider activation from words that share orthographic rhyme only

Pronunciation performance

- Word naming performance: 99.21%
- Nonword naming performance: 70.75%
- Lenient scoring criterion: non-word pronunciation is correct if the orthography-to-phonology mapping for the onset, vowel and coda exists for a mono-syllabic word in CELEX
- With lenient scoring criterion: 97.69%

Outline

- Introduction
- NDRa model
- Simulations
    - Overall model fit
    - Predictor simulations
    - Comparison to dual-route architecture
- Conclusions

Conclusions

- Discriminative learning works for reading aloud
- A single lexical route is sufficient to explain a wide range of experimental effects in both word and non-word naming

Conclusions

- Outstanding issues:
  - What is the functional architecture of the selection mechanism?
  - Discrete representations are abstractions from the neurobiological reality of language processing
  - Extension to multi-syllabic words

General conclusions

- Naive discriminative learning:
  - Competitive models of language processing
  - Based on a general learning mechanism
  - Parsimonious
  - Computationally efficient implementation in the `ndl` package

General conclusions

Thank you!