Quantitative Comparison for Generative Theories:

Embedding Competence Linguistic Theories in Cognitive Architectures and Bayesian Models

Adrian Brasoveanu & Jakub Dotlačil

BLS 44, UC Berkeley · February 9, 2018

The main goal

Introduce a new framework integrating generative theories, ACT-R models, and Bayesian methods.

i. Generative theories + ACT-R: competence-level generative theories are embedded in performance-level processing ACT-R models
(Anderson and Lebiere 1998, Lewis and Vasishth 2005 a.o.)

- this enables us to explicitly and fully model the behavior of human participants in standard experimental tasks
(lexical decision, forced-choice, self-paced reading, eye-tracking)

This is computationally implemented in a new Python3 library: **pyactr**, **`https://github.com/jakdot/pyactr`**.
(If you use this Python3 library, please cite it as Brasoveanu and Dotlačil (2018, in prep.) and include the github url.)

The main goal

ii. ACT-R + Bayes: the ACT-R models are embedded in Bayesian models; we can then fit them to experimental data and do quantitative comparison for qualitative theories

- **pyactr** enables us to easily interface ACT-R models with standard statistical estimation methods implemented in widely-used Python3 libraries
- we use ACT-R models as the likelihood component of full Bayesian models, and fit the ACT-R parameters to experimental data
- upshot: we are able to consider alternative generative grammar theories and quantitatively compare how well they fit experimental data

The main goal

The ability to do quantitative comparison for qualitative generative theories on this scale is unprecedented (as far as we know).

- even in ACT-R, subsymbolic/quantitative parameters are usually set by hand instead of estimated from the data using standard statistical estimation methods

A detailed introduction to the framework will be available soon in Brasoveanu and Dotlačil (2018, in prep.). **Today, a case study**:

- the lexical decision task in Murray and Forster (2004)
- we model their data with 3 different ACT-R models that differ qualitatively / symbolically or quantitatively / subsymbolically
- we fit these models to data and compare the results

Road map for the talk

- we introduce the lexical decision task and the data we want to model
- we discuss a basic Bayesian log-frequency model for this data; this model
  - highlights the imperfect data fit of the log-frequency assumption
  - and introduces the basic structure of a Bayesian model we will need later
- we introduce a series of 3 ACT-R models of a participant completing the lexical decision task and quantitatively compare them
- these lexical access models are particularly simple – the framework can accommodate much more realistic linguistic theories
  - (if there's time) we demo an incremental left-corner parser & interpreter (using DRT on the semantics side) with visual and motor interfaces

The lexical decision task in Murray and Forster (2004)

- word frequency: one very robust parameter affecting latencies and accuracies in lexical decision tasks (Whaley, 1978)
- frequency effects have been found in many if not all tasks that involve some kind of lexical processing (Forster, 1990; Monsell, 1991)
- specific functional form: lexical access latency can be well approximated as a log-function of frequency (Howes and Solomon 1951)
- Murray and Forster (2004) studied the role of frequency in detail and identified various issues with the log-frequency model
- their data consisted of collected responses and response times in a lexical decision task using words from 16 frequency bands – see table on the next slide

The lexical decision task in Murray and Forster (2004)

The 16 word-frequency bands (in tokens per 1 million words) investigated in Murray and Forster (2004), Exp. 1:

Frequency range	**Mean frequency**	**Latency (ms)**	Accuracy (%)
315–197	**242.0**	**542**	97.22
100–85	**92.8**	**555**	95.56
60–55	**57.7**	**566**	95.56
42–39	**40.5**	**562**	96.3
32–30	**30.6**	**570**	96.11
24–23	**23.4**	**569**	94.26
19	**19.0**	**577**	95
16	**16.0**	**587**	92.41
14-13	**13.4**	**592**	91.67
12–11	**11.5**	**605**	93.52
10	**10.0**	**603**	91.85
9	**9.0**	**575**	93.52
7	**7.0**	**620**	91.48
5	**5.0**	**607**	90.93
3	**3.0**	**622**	84.44
1	**1.0**	**674**	74.63

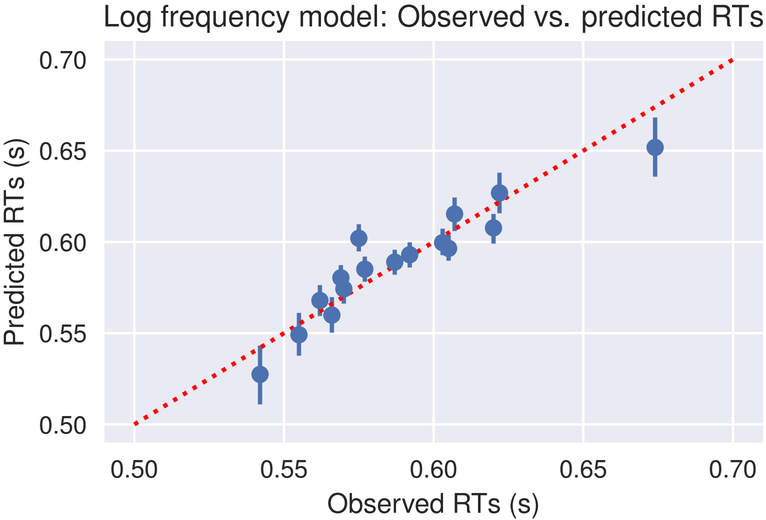
Specifying a Bayesian log-frequency model

To get acquainted with the structure of a Bayesian model, let's specify a simple Bayesian log-frequency model:

```
log_freq_model = Model()
with log_freq_model:
    # priors
    intercept = Normal(...)
    slope = Normal(...)
    # likelihood
    mu = Deterministic(intercept + slope*np.log(freq), ...)
    observed_rt = Normal(mu=mu, observed=rt, ...)
    # sample posterior
    trace = sample(draws=5000, ...)
```

The predictions of the log-frequency model

Figure: Log-frequency model estimates and observed RTs



Frequency effects as practiced memory retrieval

- log-frequency gets middle values right, but underestimates time needed to access words in extreme frequency bands
- **our proposal**: frequency effects as practiced memory retrieval

  (different from the proposal in Murray and Forster 2004)
- memory retrieval (practice and forgetting): a power function of time (Newell and Rosenbloom 1981, Anderson 1982, Logan 1990)

Frequency effects as practiced memory retrieval

- practice: repeated presentation of an item
- ACT-R: retrieval from declarative memory is a power function of time elapsed since item presentation
- the power function is used to compute (base) activation and is based on the number of practice trials / ‘rehearsals’ of a word (1) (free parameters enumerated in parentheses)
- activation of an item is in turn used to compute accuracy (2) and latency (3) for retrieval processes

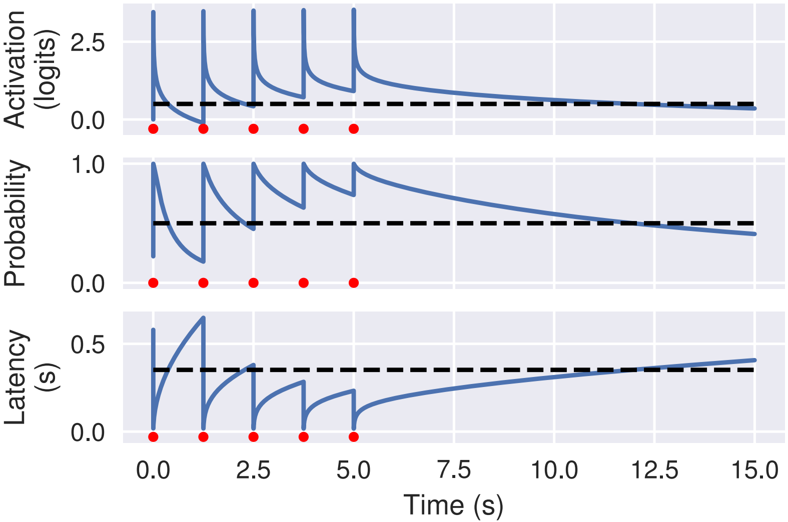
(1) $A_i = \log\left(\sum_{k=1}^{n} t_k^{-\mathbf{d}}\right)$ (**d**: decay)

(2) $P_i = \frac{1}{1+e^{-\frac{A_i-\tau}{\mathbf{s}}}}$ (**s**: noise, τ: threshold)

(3) $T_i = \mathbf{F}e^{-\mathbf{f}A_i}$ (**F**:factor, **f**: exponent)

Frequency effects as practiced memory retrieval

Figure: Activation, retrieval probability and retrieval latency as a function of time (threshold – dotted black line; 5 presentations – red)



Frequency effects as practiced memory retrieval

- for any word, the number of rehearsals that contribute to its activation are determined by its frequency (we ignore other factors in this model)
- we generate a rehearsal / presentation schedule for a 15-year old speaker based on word frequency and the average number of words the 15-year old speaker is estimated to have seen (estimate based on Hart and Risley 1995)

Bayesian model with ACT-R likelihood for RTs

Embed ACT-R models in Bayesian models to link them to data:

```
lex_decision_with_bayes = Model()
with lex_decision_with_bayes:
    # priors for model parameters
    d = ...
    s = ...
    tau = ...
    F = ...
    f = ...
    # likelihood: RTs are based on the ACT-R model
    pyactr_rt = actrmodel_latency(F, f, d, activation_from_time)
    rt_observed = Normal(mu=pyactr_rt, observed=RT, ...)
    prob_observed = ...
```

Bayesian model with ACT-R likelihood for RTs

- **pyactr_rt** on line 10 invokes an ACT-R model (we'll discuss these models presently), and runs it to generate lexical latencies for words in the 16 frequency bands
- the ACT-R model is parametrized by a latency factor **F**, a latency exponent **f**, a decay **d** and the activation for words in the 16 frequency bands **activation_from_time**, computed based on their 15-year long rehearsal schedule
- the 16 reaction time (RT) means from Murray and Forster (2004) are then assumed to be noisy realizations of the ACT-R generated RTs (line 11)
- for simplicity, we model the observed response accuracies directly (line 12), not via an ACT-R model

ACT-R models

ACT-R models embed competence theories in processing models.

- we have a qualitative/symbolic competence theory of the lexicon: lexical items have various features (their form etc.)
- we have a qualitative performance theory of what human participants actually do in a lexical decision task
  - lexical items are stored in declarative memory and have an activation that is a function of their frequency
  - participants read a form (sequence of characters) on the screen and attempt to retrieve a word with that form
- the qualitative components are implemented in ACT-R as **condition-action** pairs (production rules) stored in procedural memory
- these rules trigger a cognitive **action** if the cognitive context / mental state satisfies a range of **conditions**

Quantitative comparison for qualitative theories

Generative theories + ACT-R + Bayes enable us to do quantitative comparison for qualitative theories:

- we can implement different competence + processing models in ACT-R, and then embed these alternative ACT-R models in a Bayesian model
- we can then estimate their subsymbolic parameters and quantitatively compare these different models
- model comparison with Bayes factors can apply across the board for any kind of hybrid (quantitative & qualitative) model

  (if done responsibly ... Kass and Raftery 1995)

An ACT-R model of lexical decision (Model 1)

The model consists of 4 central rules:

1. The **"attend word"** rule takes a visual location encoded in the visual location buffer, a.k.a., the visual *where* buffer, and issues a command to the visual *what* buffer to move attention to that visual location

An ACT-R model of lexical decision (Model 1)

```
lex_decision.productionstring(name="attend word", string="""
    =g>
    state   attend
    =visual_location>
    isa     _visuallocation
    ?visual>
    state   free
    ==>
    =g>
    state   retrieving
    +visual>
    cmd     move_attention
    screen_pos =visual_location
    ~visual_location>
""")
```

An ACT-R model of lexical decision (Model 1)

2. The **"retrieving"** rule takes the visual value/content discovered at that visual location, which is a potential word form, and places a declarative memory request to retrieve a word with that form;

```
lex_decision.productionstring(name="retrieving", string="""
    =g>
    state   retrieving
    =visual>
    value   =val
    ==>
    =g>
    state   retrieval_done
    +retrieval>
    isa     word
    form    =val
""")
```

An ACT-R model of lexical decision (Model 1)

3\. and 4. The **"lexeme retrieved"** and **"no lexeme found"** rules take care of the two possible outcomes of the memory retrieval request

- if a word with that form is retrieved from memory (**"lexeme retrieved"**), a command is issued to the motor module to press the ’**J**’ key
- if no word is retrieved (**"no lexeme found"**), a command is issued to the motor module to press the ’**F**’ key

An ACT-R model of lexical decision (Model 1)

```
lex_decision.productionstring(name="lexeme retrieved", string="""
    =g>
    state   retrieval_done
    ?retrieval>
    buffer  full
    state   free
    ==>
    =g>
    state   done
    +manual>
    cmd     press_key
    key     J
""")
```

An ACT-R model of lexical decision (Model 1)

```
lex_decision.productionstring(name="no lexeme found", string="""
    =g>
    state   retrieval_done
    ?retrieval>
    buffer  empty
    state   error
    ==>
    =g>
    state   done
    +manual>
    cmd     press_key
    key     F
""")
```

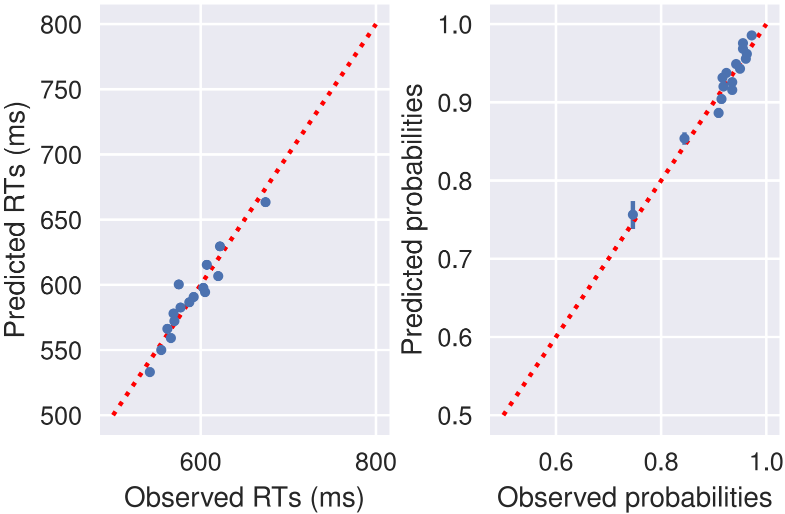
An ACT-R model of lexical decision (Model 1)

Running this model, we obtain an output detailing the cognitive process and its temporal trace:

```
****Environment: {1: {'text': 'elephant', 'position': (320, 180)}}
(0, 'PROCEDURAL', 'RULE SELECTED: attend word')
(0.05, 'PROCEDURAL', 'RULE FIRED: attend word')
(0.0679, 'PROCEDURAL', 'RULE SELECTED: retrieving')
(0.1179, 'PROCEDURAL', 'RULE FIRED: retrieving')
(0.1179, 'retrieval', 'START RETRIEVAL')
(0.1679, 'retrieval', 'RETRIEVED: word(form= elephant)')
(0.1679, 'PROCEDURAL', 'RULE SELECTED: lexeme retrieved')
(0.2179, 'PROCEDURAL', 'RULE FIRED: lexeme retrieved')
(0.2179, 'manual', 'COMMAND: press_key')
(0.4679, 'manual', 'PREPARATION COMPLETE')
(0.5179, 'manual', 'INITIATION COMPLETE')
(0.6179, 'manual', 'KEY PRESSED: J')
```

ACT-R Model 1: fit to data

Figure: Model 1: estimated and observed RTs and probabilities



ACT-R Model 1: fit to data and qualitative limitations

- the plots show Model 1 has a very good fit, both for latency and accuracy
- but Model 1 oversimplifies the process of encoding visually retrieved data
  - it assumes the visual value found at a particular visual location is immediately shuttled to the retrieval buffer
  - but cognition in ACT-R is goal-driven: any important step in a cognitive process should involve the **goal** or **imaginal** buffer
  - the **imaginal** buffer is a goal-like buffer that stores internal ‘snapshots’ of the cognitive state
- the transfer between the visual and the retrieval buffer should be mediated by the **imaginal** buffer

ACT-R Model 2: adding the **imaginal** buffer

- Bayesian model remains the same, the only part we change is the ACT-R-provided likelihood for latencies
- we modify the procedural core of the ACT-R model
  - we add the imaginal buffer to the model
  - we replace the **"attend word"** and **"retrieving"** rules with three rules **"attend word"**, **"encoding word"** and **"retrieving"**
  - the new rule **"encoding word"** mediates between **"attend word"** and **"retrieving"**
  - **encoding** a word form means taking it from the visual buffer and shuttling it to the imaginal buffer

ACT-R Model 2: adding the **imaginal** buffer

```
lex_decision.set_goal("imaginal")

lex_decision.productionstring(name="attend word", string="""
    =g>
    state   attend
    =visual_location>
    isa     _visuallocation
    ?visual>
    state   free
    ==>
    =g>
    state   encoding                        [the only change in this rule]
    +visual>
    cmd     move_attention
    screen_pos =visual_location
    ~visual_location>
""")
```

ACT-R Model 2: adding the **imaginal** buffer

```
lex_decision.productionstring(name="encoding word", string="""
    =g>
    state   encoding
    =visual>
    value   =val
    ==>
    =g>
    state   retrieving
    +imaginal>
    isa     word
    form    =val
""")
```

ACT-R Model 2: adding the **imaginal** buffer

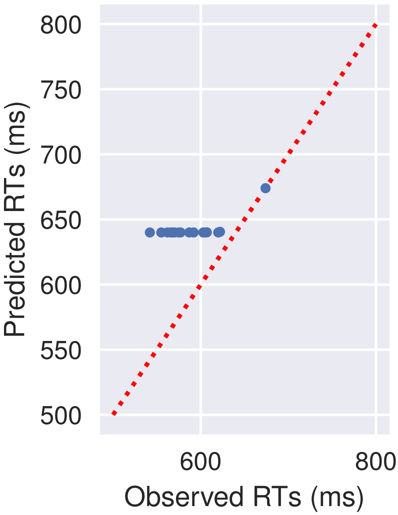
```
lex_decision.productionstring(name="retrieving", string="""
    =g>
    state   retrieving
    =imaginal>              [imaginal instead of visual: the only change in this rule]
    isa     word
    form    =val
    ==>
    =g>
    state   retrieval_done
    +retrieval>
    is      word
    form    =val
""")
```

ACT-R Model 2: adding the **imaginal** buffer

- these modifications are symbolic/discrete/non-quantitative modifications
- but we are able to fit the new model to the same data and quantitatively compare its performance with Model 1 (the no-imaginal-buffer model)
- the left plot on the next slide shows that Model 2 has a very poor fit to the latency data

ACT-R Model 2: fit to data

Figure: Model 2: estimated and observed RTs and probabilities



ACT-R Model 2: adding the **imaginal** buffer

- the encoding step adds 200 ms to every lexical decision simulation
- 200 ms is the default ACT-R delay for chunk-encoding into the imaginal buffer
- the predicted latencies for 15 out of the 16 word-frequency bands are greatly overestimated (above the diagonal line)
- Model 2 cannot run faster than about 640 ms; this is too high to fit high-frequency words, which take about 100 ms less than that

ACT-R Model 3: **imaginal** buffer with 0 delay

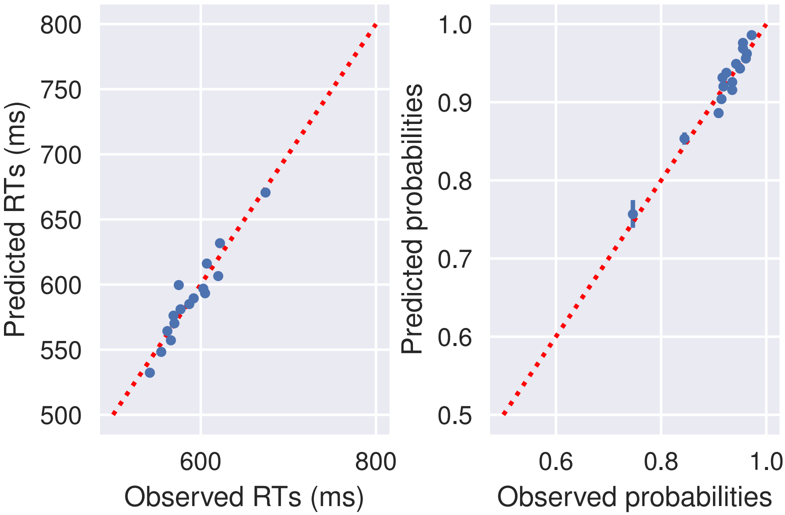
- let's change a quantitative feature of Model 2 and set the imaginal delay to 0 ms (instead of its default 200 ms value)

```
lex_decision.goals["imaginal"].delay = 0
```

- it is reasonable to assume that various default values for ACT-R subsymbolic parameters should be changed when modeling linguistic phenomena
- natural language comprehension involves fast incremental construction of rich hierarchical representations
- this richness significantly exceeds the complexity of representations needed for other high-level cognitive processes modeled in ACT-R (e.g., arithmetic)
- Model 3 fits very well the mean latencies for all the 16 word-frequency bands

ACT-R Model 3: fit to data

Figure: Model 3: estimated and observed RTs and probabilities



Conclusion

- we have a formally explicit way to connect competence-level theories to experimental data via explicit processing models
- we can formally, explicitly connect qualitative/symbolic/competence-level theory construction (the main business of the generative grammarian) and quantitative/subsymbolic/performance-level data collection and prediction (the main business of the experimental linguist)

For a future occasion – more systematic / formal model comparison:

- we have only done informal quantitative comparisons based on posterior predictions
- but systematic across-the-board model comparison via Bayes factors is possible in this framework

Demo time

An incremental left-corner parser & interpreter (using DRT on the semantics side) with visual and motor interfaces

...applied to cataphora, specifically the conditional:

(4) John won't eat **it** if **a hamburger** is overcooked. (Elbourne 2009, p. 3)

The model provides an end-to-end simulation of a human participant in a self-paced reading task (Just et al. 1982):

- it reads the conditional in (4), which is displayed one word at a time on a virtual screen
- it presses the space bar to move to the next word when the current word is integrated (parsed & interpreted)
- it implements a version of Discourse Representation Theory (DRT; Kamp 1981, Kamp and Reyle 1993) on the semantics side
- it builds the expected tree structures on the syntax side

Acknowledgments

We are grateful to Donka Farkas, Abel Rodriguez, Matt Wagers and the UCSC S-lab audience (January 2018) for comments and discussion. The usual disclaimers apply.

References I

Anderson, John R (1982). “Acquisition of cognitive skill”. In: *Psychological review* 89.4, p. 369.

Anderson, John R. and Christian Lebiere (1998). *The Atomic Components of Thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Brasoveanu, Adrian and Jakub Dotlačil (2018, in prep.). *Formal Linguistics and Cognitive Architecture: Integrating generative grammars, cognitive architectures and Bayesian methods*. Language, Cognition, and Mind (LCAM) Series. The pyactr library (Python3 ACT-R) is available here: https://github.com/jakdot/pyactr. Dordrecht: Springer.

Elbourne, Paul (2009). “Bishop Sentences and Donkey Cataphora: A Response to Barker and Shan”. In: *Semantics and Pragmatics* 2, pp. 1–7.

Forster, Kenneth I (1990). “Lexical processing”. In: *Language: An invitation to cognitive science*. Ed. by Daniel Osherson and Howard Lasnik. Cambridge, MA: MIT Press, pp. 95–131.

Hart, Betty and Todd R Risley (1995). *Meaningful differences in the everyday experience of young American children.*. Baltimore: Paul H Brookes Publishing.

References II

- Howes, Davis H and Richard L Solomon (1951). "Visual duration threshold as a function of word-probability.". In: *Journal of experimental psychology* 41.6, p. 401.
- Just, Marcel A. et al. (1982). "Paradigms and processes in reading comprehension". In: *Journal of Experimental Psychology: General* 111.2, pp. 228–238. DOI: 10.1037/0096-3445.111.2.228.
- Kamp, Hans (1981). "A Theory of Truth and Semantic Representation". In: *Formal Methods in the Study of Language*. Ed. by Jeroen Groenendijk et al. Amsterdam: Mathematical Centre Tracts, pp. 277–322.
- Kamp, Hans and Uwe Reyle (1993). *From Discourse to Logic. Introduction to Model theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: Kluwer.
- Kass, Robert E. and Adrian E. Raftery (1995). "Bayes Factors". In: *Journal of the American Statistical Association* 90.430, pp. 773–795. DOI: 10.1080/01621459.1995.10476572.
- Lewis, Richard and Shravan Vasishth (2005). "An activation-based model of sentence processing as skilled memory retrieval". In: *Cognitive Science* 29, pp. 1–45.

References III

- Logan, Gordon D (1990). "Repetition priming and automaticity: Common underlying mechanisms?". In: *Cognitive Psychology* 22.1, pp. 1–35.
- Monsell, Stephen (1991). "The nature and locus of word frequency effects in reading". In: *Basic processes in reading: Visual word recognition*. Ed. by D. Besner and G. W. Humphreys. Hillsdale, NJ: Erlbaum, pp. 148–197.
- Murray, Wayne S and Kenneth I Forster (2004). "Serial mechanisms in lexical access: the rank hypothesis.". In: *Psychological Review* 111.3, p. 721.
- Newell, Allen and Paul S Rosenbloom (1981). "Mechanisms of skill acquisition and the law of practice". In: *Cognitive skills and their acquisition*. Ed. by John R. Anderson. Hillsdale, NJ: Erlbaum, pp. 1–55.
- Whaley, Charles P (1978). "Word-nonword classification time". In: *Journal of Verbal Learning and Verbal Behavior* 17.2, pp. 143–154.