Class 14, 5/16/13: More on Model Evaluation and Productivity

1. Assignments etc.

- Seuss statistics exercise due May 21.
- For Thursday 5/23: read Jennifer Hay and Harald Baayen (2002) Parsing and productivity. *Yearbook of Morphology.* Linked from course web site.
- Talk: with us re. projects.

2. Goals for today

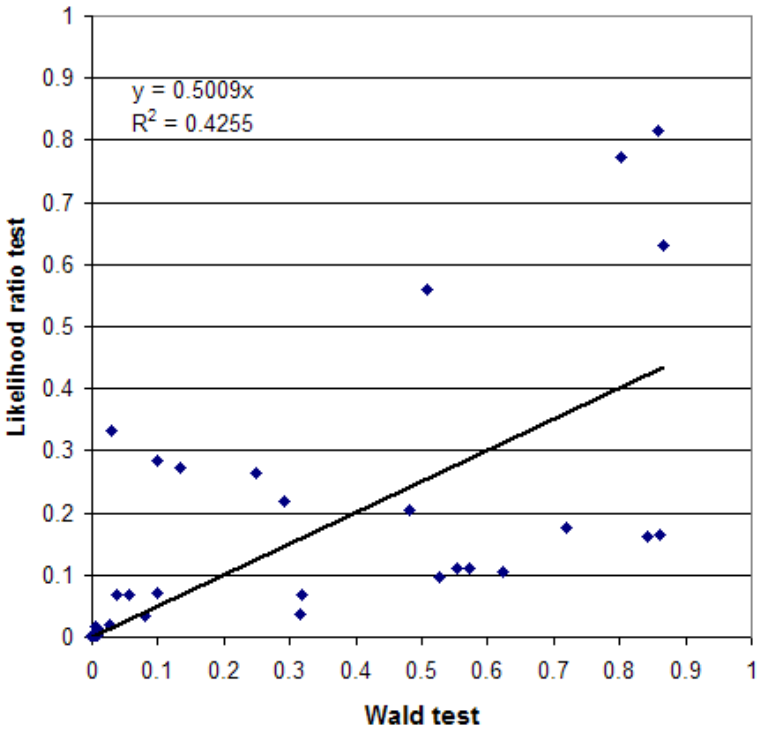
- Solidify and better understand the methods of model evaluation taught by Kie by applying them in various ways
- A survey of informal ways of assessing model overall (as opposed to individual constraints)
- Back to productivity: the ideas of Hay and Baayen

3. Four methods of model evaluation taught by Kie

- We ask: is it sensible to include a constraint in a model?
- We can inquire:
  - What is the significance figure revealed in the **Wald test** (default for logistic regression in R)?
  - What does the **Likelihood Ratio Test** tell us about the model, vs. a minimally different model with the constraint removed?
  - Does the **Akaike Information Criterion** (resp. Bayesian Information Criterion) get better (smaller) when you put the constraint in?
  - If you do **cross-validation** on the model, does including the constraint cause accuracy to drop? (overfitting)

4. Wald test vs. Likelihood Ratio Test

- Kie pointed out that one can do the Likelihood Ratio Test en masse by using this command:
  - `Anova(MyModel, type=2)`
- I tried this for the mass-Seuss model and compared the significance figures with those obtained from the Wald test.



- Findings:
  - Gaak, they're not that closely related — correlation is .662.
  - The Wald Test is in a sense more conservative — the regression line, set to pass through zero, has a slope of .43, meaning LRT is assigning lower p-values on the whole.
  - Note, however, that we don't really care much about their agreement when they both assign a non-significant interpretation — a miss is as good as a mile.
  - Let's next look at another way to assess agreement

5. Disagreements: when does one method return a significant verdict, the other not?

- $p < .05$
  - Note: this is not a good criterion to use, since we've carried out a fishing expedition for constraints

	Wald	Likelihood Ratio test
AY1	0.081	**0.034**
OY1	0.316	**0.038**
InitialGLiquid	**0.037**	0.068
AH1	**0.03**	0.333

  - Each system smirks at a "keep it" verdict rendered by the other.
- $p < .01$

FinalLabStop	0.005	0.016

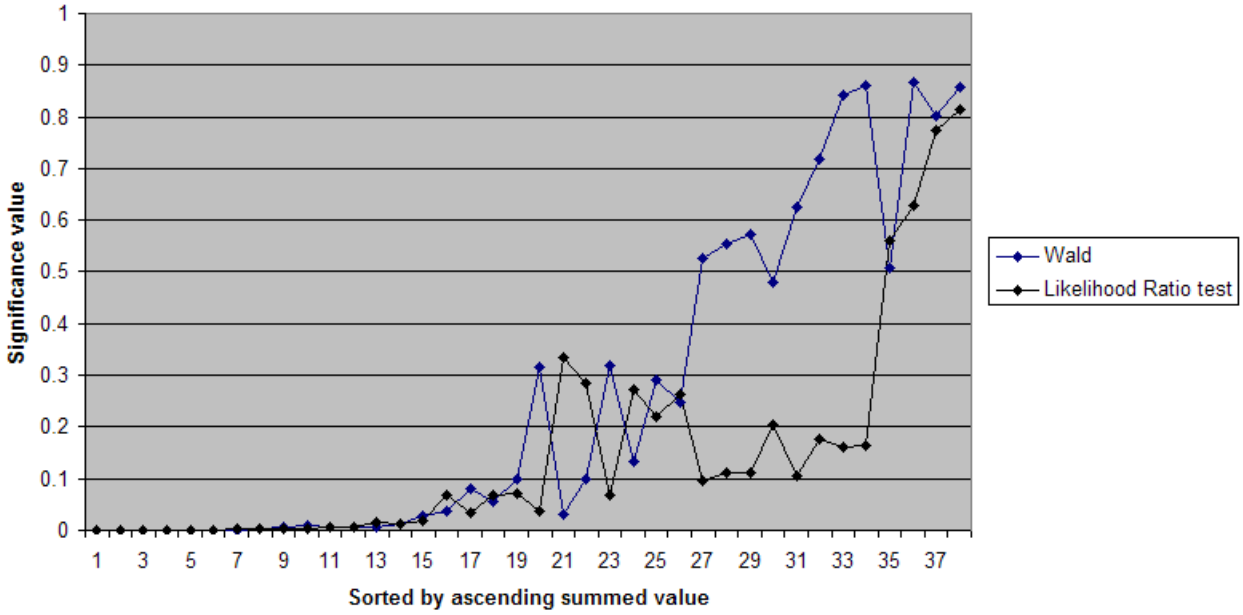
- $p < .001$

InitialFL	0	0.004

- So, closer inspection suggests that if you use a strict significance criterion, perhaps the differences between the two methods should not bother us — at least in this case.

6. Another chart suggesting the same point.

- Procedure:
  - Sum the significance values of Wald and LRT.
  - Sort by this sum, but plot each value as its own line.



  - Result: massive variation — but mostly in the (useless) upper regions.
- This is just one empirical case — we'd like to know in general when the methods are likely to differ.

7. What about catching accidentally-true constraints?

- I mentioned this in Class 5.
- Tommo So Vowel Harmony analysis, with 20 constraints that were given a completely random value for each input type.
- Some of these got fairly large weights, and passed the Wald test.

Constraint	Estimate	Std. Error	z value	Pr(>|z|)

Identlow	10.179	1.099	9.263	0
IdentATR	6.272	1.141	5.498	0
Identback	4.802	0.71	6.759	0
Agreelow	2.057	0.228	9.017	0
AgreeATR	2.387	0.343	6.951	0
Agreeback	1.354	0.235	5.75	0
Random1	-0.679	0.519	-1.308	0.191
Random2	-1.701	0.593	-2.869	**0.004**
Random3	-0.279	0.607	-0.459	0.646
Random4	-0.046	0.53	-0.087	0.930
Random5	0.095	0.814	0.117	0.907
Random6	-1.078	0.785	-1.373	0.170
Random7	-0.109	0.612	-0.179	0.858
Random8	-1.001	0.641	-1.561	0.119
Random9	-1.158	0.561	-2.065	0.039
Random10	1.234	0.866	1.426	0.154
Random11	0.456	0.837	0.545	0.586
Random12	0.985	0.553	1.782	0.075
Random13	-0.223	0.437	-0.509	0.610
Random14	1.476	0.685	2.155	**0.031**
Random15	-2.025	0.604	-3.351	**0.001**
Random16	-0.117	0.786	-0.148	0.882
Random17	0.468	0.808	0.579	0.562
Random18	0.958	0.674	1.421	0.155
Random19	-0.095	0.581	-0.163	0.871
Random20	0.797	0.564	1.413	0.158

- Could the Likelihood Ratio Test do better in flagging the random constraints as random?
- Ideally, one would do this en masse, as Kie showed last time.
- Due to R's objections (about which I worry…), I instead did the test by hand in Excel, for just Random15, the "stinker constraint".
- Result: p = .0019 — we're not any better off.
- Caveat: I now see I assigned random violations to *types* of forms, not to individual tokens. Probably the latter approach would have yielded less garbage.

MODEL EVALUATION: BEYOND INDIVIDUAL CONSTRAINTS

8. What else do we need?

- The methods just covered answer a fairly narrow question:
  - "**Is it worth it to include this constraint in my model?**"
- There are other questions we often ask:
  - "**How am I doing**?"
    = "Is it possible that I'm failing to understand this phenomenon and I should look for a different approach?"

= "Is it possible that I'm picked a bad research topic and am studying a phenomenon that is drowned in noise?"

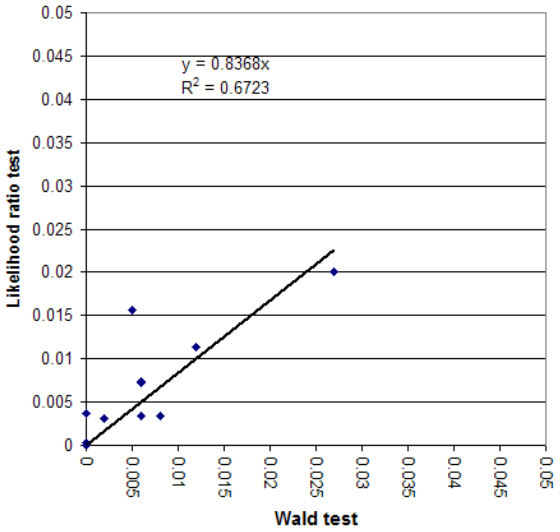
- "**How could I improve my model?**"

9. Some methods that can assist our judgment on "How am I doing?"

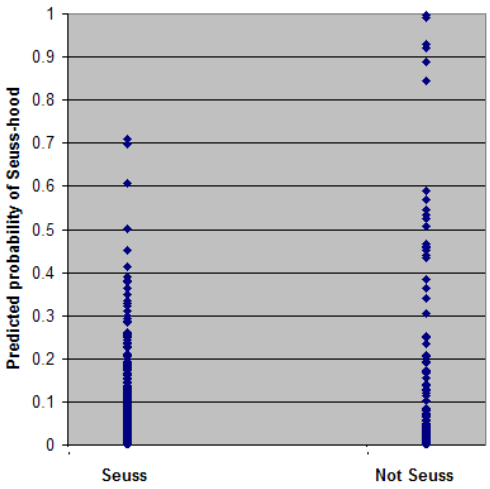
- Scattergram
- Histogram
- Correlation?
- Summed error?

10. Make a scattergram of predicted vs. observed.

- This is especially helpful when the scattergram is nicely cloud-shaped so the points don't sit on top of each other.
  - Note scattergram above: the crowding near (0,0) makes it hard to judge.
  - A zoomed-in scattergram might help:



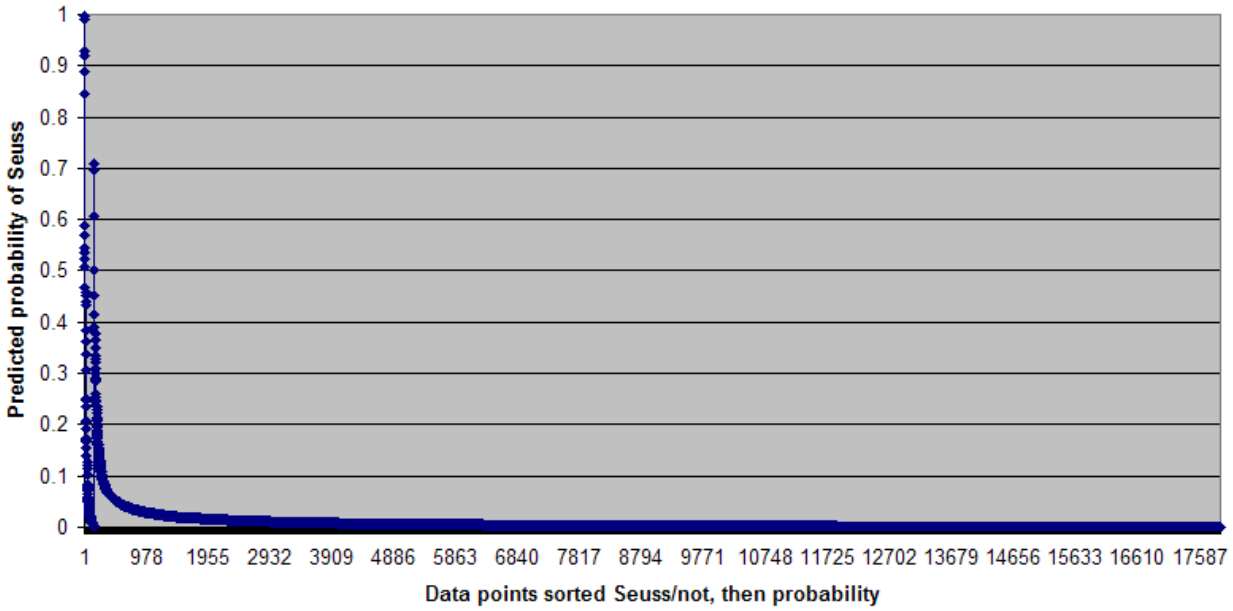
- This method won't help if the "observed" factor is a category like Seuss/Not Seuss — the categories will usually cause the points to overlap too much.
  - Here is a highly deceptive scattergram for the Seuss model:



- ➢ It's deceptive because the vast majority of the data are overlapping. 91.7% of all data are real words with predicted Seuss-itude under .02.
- ➢ In R you can make "violin plots" which thicken the vertical line according to frequency — violin probably too distorted to serve in the Seuss case

11. Histograms

- Here is one way to do it.
- In Excel, sort descending by IsSeuss, Predicted
- Make a line graph for Predicted.



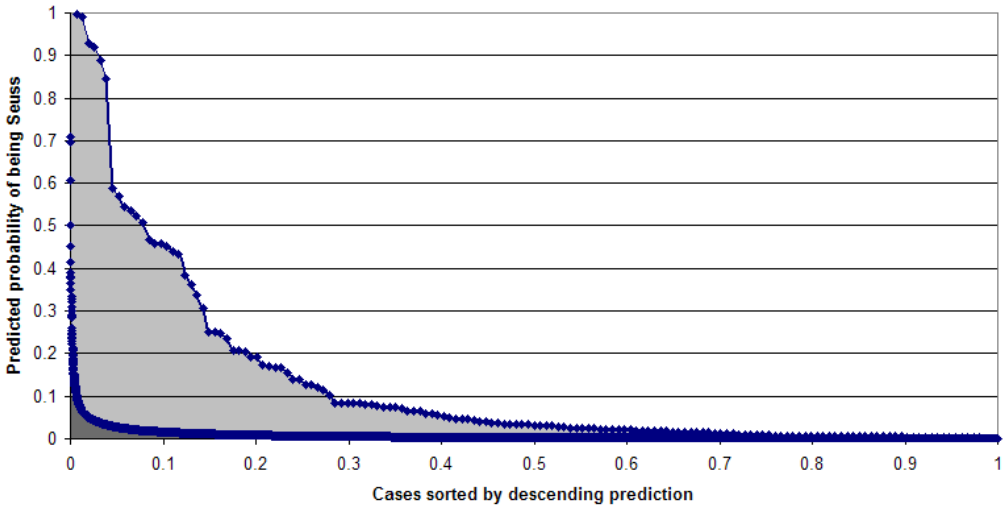
- ➢ This is not really very informative because the Seuss points are too compressed

12. Stretched histogram

- Keep the Excel sort just described.
- Make new columns in your spreadsheet that will go gradually from **zero to one** through each category.

Seuss or not	Number the instances	List total number of cases	Divide numbering by total
1	1	4	.25
1	2	4	.5
1	3	4	.75
1	4	4	1
0	1	6	.17
0	2	6	.33
0	3	6	.5
0	4	6	.67
0	5	6	.83
0	6	6	1

- Make a **scattergram** of the predicted values against the column labeled "Divide".
- For Seuss (with a bit of touch-up), this yields:



- Socrates: what color would be the diagram for a perfect model?
- Socrates: what would be the appearance of a diagram for the null model (no phonology, just the Intercept constraint[1])

[1] In substantive terms this is DON'T BE BY SEUSS.

13. Correlations

- Not necessarily a good idea, because they exaggerate model fit where there are huge numbers of zeros.
- The correlation of the mass-Seuss-model's predictions with the 1-0 indicator variable is .351.

14. Average error

- Assume a perfect model could assign a probability of 1 to every Seuss word, 0 to every non-Seuss word — zero average error.
- The average error of the mass-Seuss-model is .015.
- This is uninformative, since most forms have observed value 0 and predicted value close to zero.

15. Seeking improvement: Look at your outliers

- Find them: in Excel, for each datum compute **predicted − observed**, sort by this, and look at the top and bottom of the lists.
- This actually can be quite informative.
- We did this for the *-able* simulation, concluding that because the model was assigning identical scores to *mónitorable* and *vísitable*, we should probably add a constraint penalizing antepenultimate stress.
    - *Vísitable* had the highest error of any (overt) form in the experiment, underpredicted by 9.4%; *mónitorable* overpredicted by 4.9%

16. Outliers in a model of English phonotactics

- BLICK: LabPhon 2012 talk by me, hopefully with a published paper sooner or later.
- 192 constraints, all made up by me.
- Weighted using the UCLA Phonotactic Learner (Hayes and Wilson 2009).
- I made up little words with all *logically* possible two-segment onsets, sorting them into attested and unattested words.
- In producing the grammar, I repeatedly examined **overpenalized reals** and **underpenalized fakes**.
- Here are the most-penalized real syllable onsets. (All constraint violations penalize the onset; the rhymes were chosen to be bland.)

Wug word	
vju	5.725
θwɛt	4.009
ʃrɛt	3.936
hjut	2.646
gju	2.331
θɛt	2.076

θrɛt	2.076
fju	1.997
hwɛt	1.992
dwɛt	1.933
twɛt	1.933
bju	1.742
gwɛt	1.677

- Here are the least-penalized fakes:

srɛt	4.495	ʃlɛt	3.936
mrɛt	4.359	ʃmɛt	3.936
nrɛt	4.359	ʃnɛt	3.936
vlɛt	4.099	ʃpɛt	3.936
vrɛt	4.099	ʃtɛt	3.936
nwɛt	4.078	hrɛt	3.881
ðɛt	3.996	mlɛt	3.833
ʃkɛt	3.936	sfɛt	3.412

  - Socrates: which of these should be of less concern to us?
- This particular model was tweaked for many hours using the method described, and I've mostly run out of ideas for how to improve it.
- If pressed, I'd say: get rid of [vj] as an onset and play with having a [ju] diphthong.

17. Similar outliers for the large Seuss grammar

- Ten *most* Seussian Seuss words with their probabilities:

Zinzibar-Zanzibar	[Z IH1 N Z AH0 B AA2 R]	0.998
Gluppity-Glupp	[G L AH2 P AH0 T IY0 G L AH1 P]	0.990
Yuzz-a-ma-Tuzz	[Y AH1 Z AH0 M AH0 T AH2 Z]	0.930
Fizza-ma-Wizza-ma-Dill	[F IH2 Z AH0 M AH0 W IH2 Z AH0 M AH0 D IH1 L]	0.920
Schloppity-Schlopp	[SH L AA2 P AH0 T IY0 SH L AA1 P]	0.888
Motta-fa-Potta-fa-Pell	[M AA2 T AH0 F AH0 P AA2 T AH0 F AH0 P EH1 L]	0.846
Zuff	[Z AH1 F]	0.590
Zuk	[Z AH1 K]	0.570
Ziffer-Zoof	[Z IH1 F ER0 Z UW2 F]	0.545
Thneed	[TH N IY1 D]	0.534

- Ten *least* Seussian Seuss words:

Hinkle(-Horn)	[HH IH1 NG K AH0 L]	0.003
Fa-Zoal	[F AH0 Z OW1 L]	0.003

Katroo	[K AH0 T R UW1]	0.003
Swomee-swans	[S W OW1 M IY0]	0.003
Soobrian	[S UW1 B R IY0 AH0 N]	0.003
Nazzim	[N AE1 Z AH0 M]	0.002
Lerkim	[L ER1 K IH0 M]	0.002
Dake	[D EY1 K]	0.002
Fotichee	[F AA1 T AH0 CH IY0]	0.001
Palooski	[P AH0 L UW1 S K IY0]	0.001

BH opinion: these words are *outlandish*! We're still missing something …

- Ten *most* Seussian non-Seuss words with their probabilities:[2]

abracadabra	[AE2 B R AH0 K AH0 D AE1 B R AH0]	0.710
zigzag	[Z IH1 GCoda Z AE2 GCoda]	0.697
zaire	[Z AY2 IH1 RCoda]	0.697
zip	[Z IH1 PCoda]	0.607
snub	[S N AH1 BCoda]	0.502
xerox	[Z IH1 R AA2 KCoda SCoda]	0.452
zap	[Z AE1 PCoda]	0.414
zoom	[Z UW1 MCoda]	0.391
smooth	[S M UW1 DHCoda]	0.380
snoop	[S N UW1 PCoda]	0.378

- Ten *least* Seussian non-Seuss words with their probabilities:

trepidation	[T R EH2 P IH0 D EY1 SH AH0 N]	2.56E-05
contemplation	[K AA2 N T AH0 M P L EY1 SH AH0 N]	2.25E-05
comprehensibility	[K AA2 M P R IY0 HH EH2 N S AH0 B IH1 L AH0 T IY0]	2.23E-05
inflationary	[IH0 N F L EY1 SH AH0 N EH2 R IY0]	1.97E-05
premeditation	[P R IY0 M EH2 D AH0 T EY1 SH AH0 N]	1.91E-05
capitalization	[K AE2 P IH0 T AH0 L IH0 Z EY1 SH AH0 N]	1.89E-05
complacency	[K AH0 M P L EY1 S AH0 N S IY0]	1.86E-05
collaborationist	[K AH0 L AE2 B ER0 EY1 SH AH0 N IH0 S T]	1.53E-05
strangulation	[S T R AE2 NG G Y AH0 L EY1 SH AH0 N]	1.52E-05
industrialization	[IH0 N D AH2 S T R IY0 AH0 L IH0 Z EY1 SH AH0 N]	1.36E-05

  - These suggest why mere length is mildly anti-Seussian.

18. Side note on the Seuss exercise

- Seuss sometimes converts real words to usage as animal/place/etc. names.
- These seem to have fairly high Seuss scores, though in weighting the grammar these words were treated as non-Seuss.

  Zed [Z EH1 D] 0.136

[2] The prevalence of product/company names suggest that our colleagues in the advertising business have rather good phonesthetic intuitions …

Gitz	[G IH1 T S]	0.121
Snookers	[S N UH1 K ER0]	0.055
bloop	[B L UW1 P]	0.054
Vroom	[V R UW1 M]	0.052
Chugg	[CH AH1 G]	0.043
Didd	[D IH1 D]	0.040
Spritz	[S P R IH1 T S]	0.030
Um	[AH1 M]	0.029
Jeers	[JH IH1 R Z]	0.024
Dawf	[D AO1 F]	0.023
Beers	[B IH1 R Z]	0.020
Gack	[G AE1 K]	0.020
Dofft	[D AO1 F T]	0.014
clop	[K L AA1 P]	0.014
Bopps	[B AA1 P S]	0.014
Wog	[W AO1 G]	0.013
Frumm	[F R AH1 M]	0.012
Offt	[AO1 F T]	0.011
Tidder	[T IH1 D ER0]	0.009
Jedd	[JH EH1 D]	0.009
Natch	[N AE1 CH]	0.008
Spazz	[S P AE1 Z]	0.007
Krox	[K R AA1 K S]	0.007
Gekko	[G EH1 K OW0]	0.007
Nerd	[N ER1 D]	0.005

HAY AND BAAYEN ON PRODUCTIVITY

19. So far

- We've been playing with models that can describe productivity.
  - Kie lecture 11: models that make a choice among alternatives
  - Bruce lecture 12: models that include the Null Parse, allowing rejection

20. Derivational morphology

- Here, gaphood becomes much more prominent than with inflection.
- With some affixes, gap is the default expectation.
- cf. deadjectival noun-forming *-th*: ***greenth***
- Borrowed, or archaic, affixes are often quite unproductive

21. The mystery of productivity

- Aronoff (1976) called it "one of the "central mysteries of word-formation"

- Unlike for other cases of mysteriously-acquired knowledge, we cannot appeal to the genome.
- There seems to be no alternative but to work with frequency.
  - There is something about the *numbers* in the ambient learning data that is telling the child what morphological processes are productive.
  - What are these numbers, and how do they translate into productivity?

22. This is actually very central to grammar

- Derivational morphology sometimes seems like a trivial sideshow of the main grammar.
- But consider that syntactic operations like causative, dative shift, passive are often heavily lexicalized (He gave/*donated the library a million dollars).
- The same considerations we will see presumably hold true when a child tries to figure out if a verb can appear in a causative or double-object construction.

23. Batting-average approaches to productivity

$$\frac{\text{rule success}}{\text{rule opportunities}}$$

- This was proposed by Aronoff, early on.
- Albright and Hayes in their work call this **raw reliability**

24. Refining the batting average a bit

- Statistical lower-confidence on raw reliability = **adjusted reliability**
- Mikheev (1997), adopted by Albright and Hayes

25. Hay and Baayen's insight

- Often, a derived form clearly would merit its own lexical entry: *argument*.
- *Argument* is somehow "self-justified"; it serves as very feeble evidence that *-ment* nouns can be derived from verbs.
- Intuitively (but not precisely): their approach *filters* the data

26. How could we justify that *argument* has its own entry?

- Semantic non-compositionality (it doesn't really mean "the activity of arguing")
- Frequency: hear a "derived form" a lot — e.g. more than its base — and there isn't much point to deriving it.[3]
  - So maybe it's the **relative frequency** of base and derived form that is used to determine whether the derived form is evidence for productivity.

[3] More precisely, we recognize the bimorphemicness of *argument* but we somehow do not *depend* on it.

27. Concrete examples from Hay and Baayen

- Tell me how you feel about them.

Base	Google hits	Derived	Hits	Ratio
arrest	88,500,000	*arrestment*	98,400	0.001
dazzle	20,600,000	*dazzlement*	101,000	0.005
pronounce	16,700,000	*pronouncement*	2,360,000	0.14
pave	21,500,000	*pavement*	26,300,000	1.22
argue	76,900,000	*argument*	221,000,000	2.87
assess	85,400,000	*assessment*	241,000,000	2.82
govern	51,000,000	*government*	1,050,000,000	20.6

28. A processing point of view (Hay and Baayen, speculating)

- You hear me say *the mauveness of this fabric impressed me.*
- You have no alternative but to inwardly apply your word-formation rule for *-ness* if you want to understand me.
- So your rule "gets some exercise", gets some credibility.

29. Cashing out the core idea with data-crunching

- We need both
  - things to count to predict productivity
  - practical measures of productivity

30. Practical measures of productivity

- Baayen and colleagues like to use the **hapax measure**.
  - Baayan, Harald (1991) "Quantitative aspects of morphological productivity," *Yearbook of Morphology* 1991, 109-149.

31. Defn.

hapax legomenon = "hapax"
= a form that occurs just once in a corpus

- Old Greek term is used because hapaxes are of great concern to the editors of Classical and Biblical texts.

32. Baayen's productivity metrics

P	total hapaxes with this affix / total tokens with this affix
*P**	total hapaxes with this affix / total hapaxes
V	total word types formed with the suffix

33. How to interpret the metrics

- Probably-silly way: the child attends to them in order to learn about productivity.
  - This seems like the tail wagging the dog; these are not robust data.
- Perhaps-sensible way: simply use them as diagnostics.
  - A productive affix will be constantly used in novel circumstances, littering the corpus with hapaxes.
  - For an unproductive affix, all the listed forms will get uttered occasionally, and eventually they will all de-hapaxify themselves in the corpurs.

34. Notes

- These measures can only make intercorpus comparisons.
- If we had a corpus consisting of every English word ever written or spoken, there would be very many words and very few hapaxes—saturation.
- But of course the individual speaker has no access to this corpus.

35. Some affixes thus rated on a large corpus

- Source: Baayan, Harald (1991) "Quantitative aspects of morphological productivity," *Yearbook of Morphology* 1991, 109-149.

  - 18,000,000 words of English:

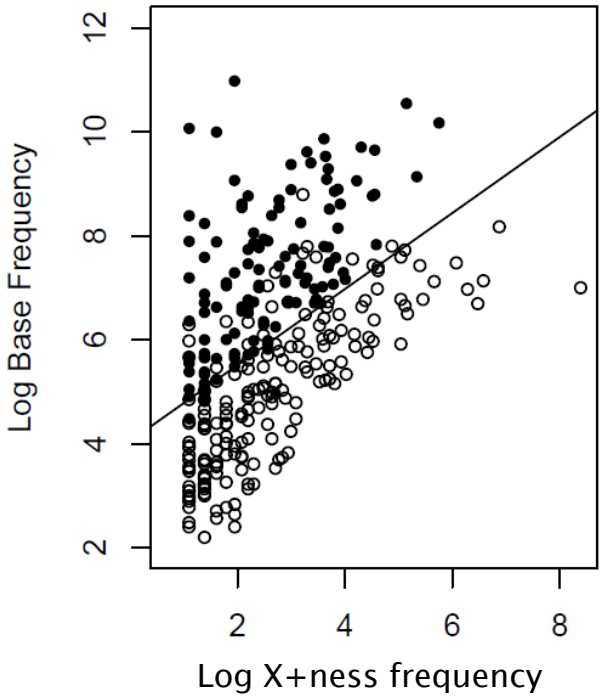
Category	Tokens	Types	Hapaxes	Hapaxes/Corpus Size
-ity	42,000	405	29	.0007
-ness	17,000	497	77	.0044
Monomorph.	2,100,000	5543	128	.0001

  - The idea is that monomorphemes can't be "productively formed" at all, so they form a kind of floor of productivity.
  - Comparable results for two semantically similar Dutch suffixes, *-te* and *-heid*

36. How to assess whether words are productively parsed?

- Hay and Baayen resort to experimental psycholinguistics and modeling.
- It's widely believed that when you hear, e.g. *happiness*, that the listed entry for the full words competes with the rule that detects *-ness* and peels it off.
  - You can study this with, e.g. priming experiments.
- Matcheck, a computational model, makes predictions — that work pretty well — about whether the rule-based candidate or the listed candidate will win.
  - It uses lexical frequency to make these predictions.

37. Locating the Parsing Line



- White vs. black dots are an independent source of quasi-data: does Baayen and Scheuder's "Matcheck" parsing model return
  - "bimorphemic" (white)
  - or "monomorphemic" (black)?
- Note that the line is *not* at y = x.
  - I.e. an affixed form, to be felt to be affixed, must be quite a bit rarer than equality with the stem.

38. The grand strategy

- Use the parsing model to find where to put the Parsing Line.
- Use the Parsing Line to predict the general productivity of particular affixes.
- Check this predicted general productivity with hapax evidence.
  - and, ideally, ultimately, with human judgements?

39. Using the parsing line

- Find the parsing line for the affix you wish to study.
- Find the fraction of forms that are above the parsing line.
- This is one form of productivity: it is predictive of the P metric given in (32) above; i.e. "total hapaxes with this affix / total tokens with this affix"
- For the second form of productivity: study all the affixes. Then we get the P* metric of (32).: "P* – the likelihood, given all productively coined words, that a coined word will contain the affix of interest, is a function of the frequency of activation of that affix – as

measured by the *number of forms containing the affix which tend to be accessed via parsing.*"

40. If they are right, what remains to be done?

- Development of a full generative model of productivity — e.g. one that could take contextual phonological influences into account.
- Development of a learning model for productivity — one that achieves Hay and Baayen's descriptive patterns by following some sort of algorithm.