Psych229: Language Acquisition

Lecture 16

Productivity - Modeling

Yang (2005): Productivity

Rules & Exceptions

Linguistic knowledge is comprised of lots of rules - but there are exceptions, too

Ex: Morphology

kiss-kissed, dance-danced, ….sing-sang

king-kings, goblin-goblins,…child-children, dwarf-dwarves

Chomsky & Halle, 1968: "…existence of exceptions does not prevent the systematic formulation of those regularities that remain"

How does a child extract the regularity that's there?

Big question: How does a child know what's systematic/productive?

Yang (2005): Productivity

Built-In learning component that recognizes productivity

productive!

Mathematical formulation = Tolerance Principle

Recognizes a productive process (way of defining what is productive from the child's perspective).

Dealing with noise in the data

If something isn't productive, can just memorize it - rather than trying to account for it in the grammar

Test case: English past tense morphology

kiss-kissed, dance-danced, …., sing-sang, go-went, make-made

Need a way to decide which rules are productive

rule 1: "+ed" (kiss-kissed)

rule 2: no change (cut-cut)

…

Yang (2005): Productivity

The errors kids make with the past tense

I holded the rabbit

Most are over-regularizations: hold-holded (make up 10% of all irregular past tense forms: Marcus et al. 1992; Yang 2002)

Very rare are over-irregularizations: bring-brang (0.2% of irregular past tense forms: Xu & Pinker, 1995)

Cross-linguistically: most errors are over-regularizations or omissions of past tense morphology (Phillips 1995; Guasti 2002)

The point: "Children recognize and generalize productive rules while memorizing the restricted use of unproductive ones"

Yang (2005): Productivity

Some definitions

Default: "when all else fails"

When more specific rules fail to apply, use this rule (which by definition is the most general).

English past tense:
+ed
kiss-kissed

Productive: "predictable" or "generalizable"

A rule automatically applies to a set of lexical items characterized by a certain context. It can extend to novel items that fit this context (though may not always)

Possible hypothesized rule:
If a verb is monosyllabic and ends in -ing, change to -ang

sing-sang,
spling-splang/splinged

A default rule is always productive, but a productive rule can exist without being the default. Neither kind of rule needs to be exception-less.

Yang (2005): Productivity

First: How do kids learn what rules there are in the first place?

Not about just looking at the rule that applies to the majority of the word types

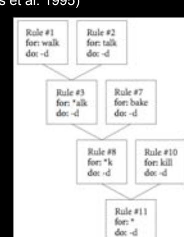
German noun plurals have a default +s that appears in only 7% of the words

Wind-Winde, Kind-Kinder, Frau-Frauen, Daumen-Daumen, Auto-Autos

(Marcus et al. 1995)

Sussman-Yip model of Molnar (2001): About looking at the context where the rule applies

Learning +ed rule (default that applies without restrictions)



Yang (2005): Productivity

Then: How do kids learn what rules are productive and what ones only apply to a restricted set (morpholexical)?

Productivity of a rule depends on knowledge of current items it applies to.

ring-rang….sing-sang… → ing-ang is productive!
…
bring-brought…sting-stung… → Not so productive once more items are encountered…

Sussman-Yip child

Point: Productivity of rule depends on some kind of cost-benefit analysis, given the items that follow the rule and the items that don't.

Yang (2005): Productivity

Idea: Cost-benefit analysis based on computational complexity

Empirical evidence points to time complexity as a sensible metric - how long does it take to access the right rule? (Morphological processing is oriented towards time efficiency.)

Question: What is the threshold for determining if a rule is productive or not?

We want some way a child could calculate this, some algorithm based on the time it takes to access the correct rule. This is what the Tolerance Principle is supposed to do.

The computational process of morphologically derived words: executed sequentially (Carmazza 1997; Levelt et al. 1999)

1) Word search (look up the word stem in the lexicon: *dance*)
2) Rule selection (find the right rule to use: *dance + ed*)
3) Rule application (apply the rule to get the derived form: *danced*)

Productivity Assessment/Tolerance Principle deals with this part

Yang (2005): Productivity

Rule selection: Lexical Search Theory (Rubenstein et al. 1970; Forster 1976)

Lexical processing involves serial search that is sensitive to the token frequencies of the words.

Idea: Rule selection also involves serial search, listed by token frequency.

Elsewhere Condition Serial Search (ECSS)
Rule: *ing-*ang

If word = *sting* then *stung* (freq 100)
Else if word = *swing* then *swung* (freq 80)
Else if word = *ding* then *dinged* (freq 10)
Else if word = *cling* then *clung* (freq 8)
Else Apply *ing --> *ang

Yang (2005): Productivity

Rule selection: Lexical Search Theory (Rubenstein et al. 1970; Forster 1976)

Lexical processing involves serial search that is sensitive to the token frequencies of the words.

Idea: Rule selection also involves serial search, listed by token frequency.

Elsewhere Condition Serial Search (ECSS)
Rule: *ing-*ang

If word = *sting* then *stung* (freq 100)
Else if word = *swing* then *swung* (freq 80)
Else if word = *ding* then *dinged* (freq 10)
Else if word = *cling* then *clung* (freq 8)
Else Apply *ing --> *ang

swing?
--> swung

Time units: 2

Yang (2005): Productivity

Rule selection: Lexical Search Theory (Rubenstein et al. 1970; Forster 1976)

Lexical processing involves serial search that is sensitive to the token frequencies of the words.

Idea: Rule selection also involves serial search, listed by token frequency.

Elsewhere Condition Serial Search (ECSS)
Rule: *ing-*ang

If word = *sting* then *stung* (freq 100)
Else if word = *swing* then *swung* (freq 80)
Else if word = *ding* then *dinged* (freq 10)
Else if word = *cling* then *clung* (freq 8)
Else Apply *ing --> *ang

ring?
--> rang

Time units: 5+rule application

Yang (2005): Productivity

Trade off: Storing individual exceptions + rules vs. exceptions only

If there are few enough exceptions, then it's more efficient to store the exceptions and then have the rule as an "elsewhere" options.

Elsewhere Condition Serial Search (ECSS)
Rule: *ing-*ang

If word = *sting* then *stung* (freq 100)
Else if word = *swing* then *swung* (freq 80)
Else if word = *ding* then *dinged* (freq 10)
Else if word = *cling* then *clung* (freq 8)
Else Apply *ing --> *ang

If there are too many exceptions, then it's more efficient to store the exceptions alone and not have a rule.

Elsewhere Condition Serial Search (ECSS)
*Rule: *ake-*ade (make-made)*

If word = *bake* then *baked* (freq 600)
Else if word = *take* then *took* (freq 400)
Else if word = *shake* then *shook* (freq 200)
Else if word = *rake* then *raked* (freq 100)
…
Else if word = *slake* then *slaked* (freq 1)

Yang (2005): Productivity

Tolerance Principle: How many is too many exceptions?

N = number of items that fit the context the rule applies to
M = number of items that are exceptions to the rule

T(M, N) = time it takes to find out if a rule applies to a given word when there are M exceptions and N items that have the rule's context

T(N,N) = time it takes to find out if a rule applies to a given word when all words are stored as exceptions

When it takes longer if exceptions are stored along with a rule (T(M,N)) than it does if all words are stored as exceptions (T(N,N)), don't bother storing the rule. The rule is not productive.

If T(N,N) < T(M,N), rule is not productive. Don't store rule.
(This happens when M ≈ N/ln N)

Yang (2005): Productivity

Tolerance Principle: Main Idea

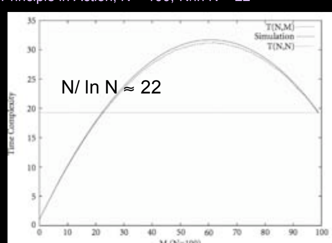
If the child knows a rule whose context fits N words, the child should only store the rule explicitly if the number of exceptions M is less than N/ln N. Otherwise, the child should store the words the rule applies to on an individual basis.

M <= N/ln N

M >= N/ln N

Yang (2005): Productivity

Tolerance Principle in Action, N = 100, N/ln N = 22



If more than about 22 words are exceptions, then it's faster to just store all the words as exceptions (because 78 words have to wait 22 time units before the rule can be applied).

Yang (2005): Productivity

Tolerance Principle predictions for English past tense morphology

Default +ed rule can only be productive if it applies to the vast majority of types it could apply to. There are 150 irregular verbs (M=150), so there need to be at least 1000 regular verbs (N=1000) for it to be faster to have a rule + exceptions. This seems to be true (we have a lot of regular verbs).

Tolerance Principle for children learning

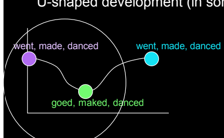
1) Child identifies possible rule. (*ing --> *ang)
2) Child (unconsciously) checks current vocabulary with Tolerance Principle to see if it's better to store a rule + exceptions, or just exceptions.
3) Child repeats with each new word type encountered. (Productivity of rules can change.)

Yang (2005): Productivity

Tolerance Principle predictions for child acquisition

By the time the child has a productive rule (like +ed), the child should know a good deal more regular verbs than irregular verbs. This seems to be true (Marcus et al. 1992).

U-shaped development (in some children) - or at least the initial dip:



1) Initially, irregular verbs learned first because they're frequent.
2) Only a few regular verbs required to posit +ed rule (20-30).
3) At this point, kids may have rule but it may not be productive because they haven't learned enough regulars. (Too many exceptions.) [initial stage]
4) Once they do see enough (M < N/ln N), then they use the rule productively. [dip of U-curve]

U-shape based solely on child's vocabulary input (how many exceptions they're exposed to)

Yang (2005): Productivity

Tolerance Principle predictions for English plural nouns

English plural nouns: Many regular nouns initially, few irregulars.
+s rule (goblin-goblins) becomes productive very quickly. No initial good performance with irregulars.
Should never see U-shaped curve in development - only an increase in performance. This seems to be true (Brown 1973, Falco & Yang 2005)

Tolerance Principle predictions for German plural nouns

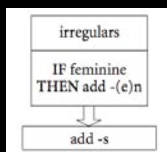
German plural nouns: many "irregular" regular rules
Ex: +en for feminine nouns (Frau - Frauen)

M = 80 exceptions
Tolerance Principle predicts at least N where N/ln N >= 80 to have a productive rule.
There must be N = 500 feminine nouns (and 420 that follow the +en rule).
There are at least 3600.

Therefore, this rule should be productive (and seems to be): Wiese 1996; Dressler 1999; Wunderlich 1999

Yang (2005): Productivity

Process for forming the plural in German



More context-specific rule (feminine)

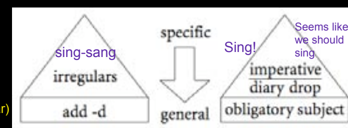
Elsewhere condition rule

Yang (2005): Productivity

Elsewhere condition in morphology & syntax

Context-specific
(not default pattern)

Elsewhere
(general core grammar)



dance-danced

We should sing