Midterm2 Review

Types of Grammars

- Type 0 Grammars *Recursively Ennumerable*
  - No restrictions on rules: rules may be recursive, and any number of symbols may occur on either side of a rule
- Type 1 Grammars *Context-Sensitive Grammars*
  - Grammars in which every rule is of the form

    $\sigma A \tau \rightarrow \sigma \phi \tau$
  - Where A is nonterminal and σ and τ are arbitrary strings of terminals and nonterminals, with ϕ nonempty

Types of Grammars

- Type 2 *Context-Free Grammars*
  - Grammars in which every rule is of the form

    $A \rightarrow \phi$
  - Where A is a nonterminal and ϕ is an arbitrary nonempty string of terminals and nonterminals
- Type 3 *Finite State Grammars*
  - Grammars in which every rule is of the form

    $A \rightarrow xB$ or $A \rightarrow x$
  - A and B are single nonterminals
  - x is an arbitrary string of terminals

Intuitions

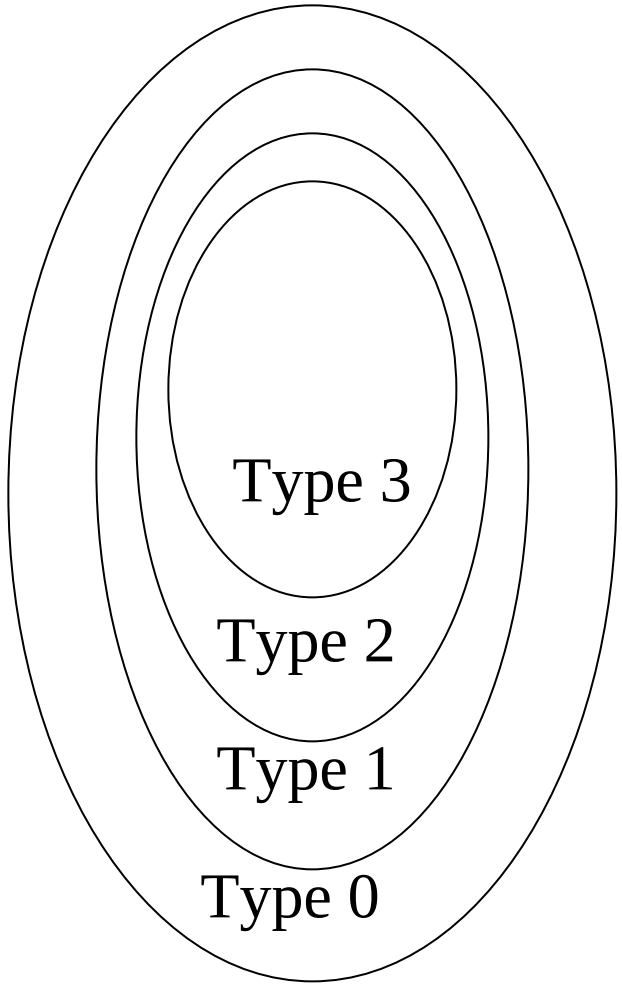
- Type 1 and Type 2 Grammars
  - Sentences made up of phrases
  - Phrases made up of smaller phrases
- Type 2
  - A Prep Phrase
  - ϕ in the doghouse
- Type 1
  - Certain types of phrases differ in different grammatical environments

  NP VP → N Det VP

  V NP → V Det N
- Type 3
  - Generate sentences left to right

Relationships between Languages

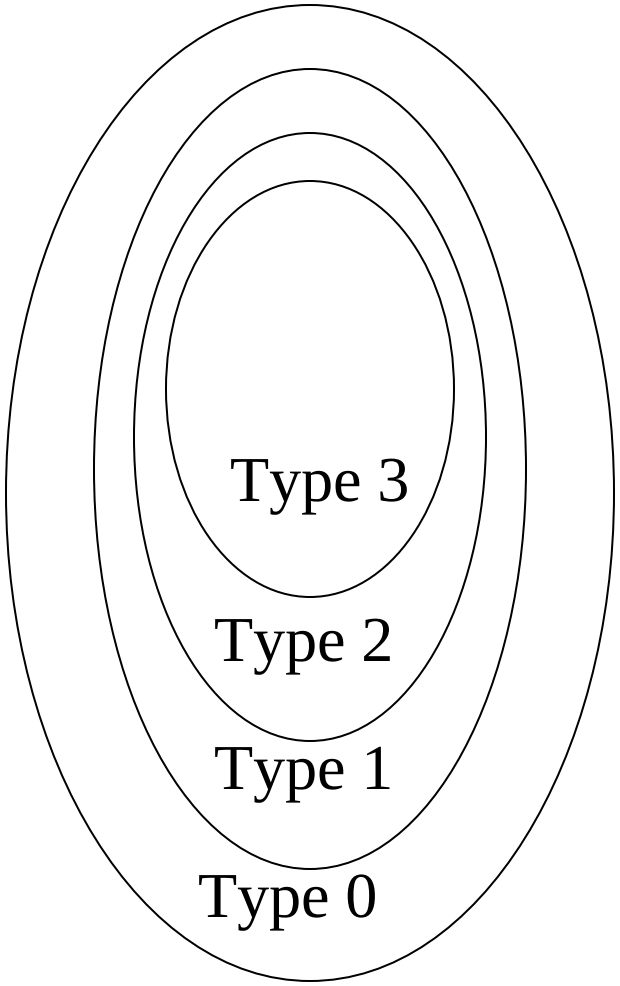
- Type 0 grammars with rules of equivalent length on the left & right sides generate all the Type 1 languages
  - Type 1 languages a subset of Type 0 languages
- Type 1 grammars in which σ and τ are always empty generate all the Type 2 languages
  - Context-Sensitive versus Context-Free
  - Type 2 subset Type 1



Automata

- Turing Machine
  - Infinite tape!
- Linear Bounded Automata
  - Available memory is a linear function of length of input
- Pushdown Automata
  - Stack memory with last in first out pattern
- Finite State Automata
  - No internal memory

Chomsky Hierarchy



- Type 0
  - Recursively Enumerable Grammar
  - Turing Machine
- Type 1
  - Context-Sensitive Grammar
  - Linear Bounded Automata
- Type 2
  - Context Free Grammar
  - Pushdown Automata
- Type 3
  - Finite State Grammar
  - Finite State Automata

Your turn!

Which is more complex?

- Type 0 grammar or Type 1 grammar?
- Recursively enumerable language or context-sensitive language?
- Type 1 or Type 2?
- Context-sensitive or context-free?
- Type 2 or Type 3?
- Context-free or finite state?
- Type 0 or Type 3?
- Recursively enumerable or finite state?
- Type 1 or Type 3?
- Context-sensitive or finite state?
- Type 0 or Type 2?
- Recursively enumerable or context-free?

Which is a subset of which?

- Type 1 languages and Type 0 languages?
- Context-sensitive languages and recursively enumerable languages?
- Type 2 languages and Type 1 languages?
- Context free languages and context sensitive languages?
- Type 3 languages and Type 2 languages?
- Finite state languages and context-free languages?
- Type 1 languages and Type 3 languages?
- Context-sensitive languages and finite state languages?
- Type 0 languages and Type 2 languages?
- Recursively enumerable languages and context-free languages?
- Type 2 languages and type 3 languages?
- Context-free languages and finite state languages?

What kind of automaton do you need to recognize a

- Type 0, recursively enumerable language?
- Type 1, context-sensitive language?
- Type 2, context-free language?
- Type 3, finite state language?

Could a Turing machine recognize

- Type 0, recursively enumerable language?
- Type 1, context-sensitive language?
- Type 2, context-free language?
- Type 3, finite state language?

Could a finite state automaton recognize

- Type 0, recursively enumerable language?
- Type 1, context-sensitive language?
- Type 2, context-free language?
- Type 3, finite state language?

Could a linear bounded automaton recognize a

- Type 0, recursively enumerable language?
- Type 1, context-sensitive language?
- Type 2, context-free language?
- Type 3, finite state language?

Could a pushdown automaton recognize

- Type 0, recursively enumerable language?
- Type 1, context-sensitive language?
- Type 2, context-free language?
- Type 3, finite state language?

What can a finite state automaton do?

L1: $a^n b^n$ n>=1

Can FSA handle this?

ab

aabb

aaabbb

*aab

*abbb

How to generate $a^n b^n$

- S → aSb
- S → ab

aSb

aabb

aSb

aaSbb

aaaSbbb

etc.

Let’s draw some trees for sentences in L1

- S → aSb
- S → ab
- ab
- aabb
- aaabbb

Embedded English Sentences (Dependencies)

- Daddy, what did you bring that book I don't want to be read to out of up for?
- How Ann can claim Pam Dawber's anger at not receiving her fair share of acclaim for Mork and Mindy's success derives from a fragile ego escapes me.

- *Can these structures be generated with a finite state grammar and parsed with a finite state automaton? Why or why not?*

Draw 2 possible trees for

- The old man in the chair with the broken leg

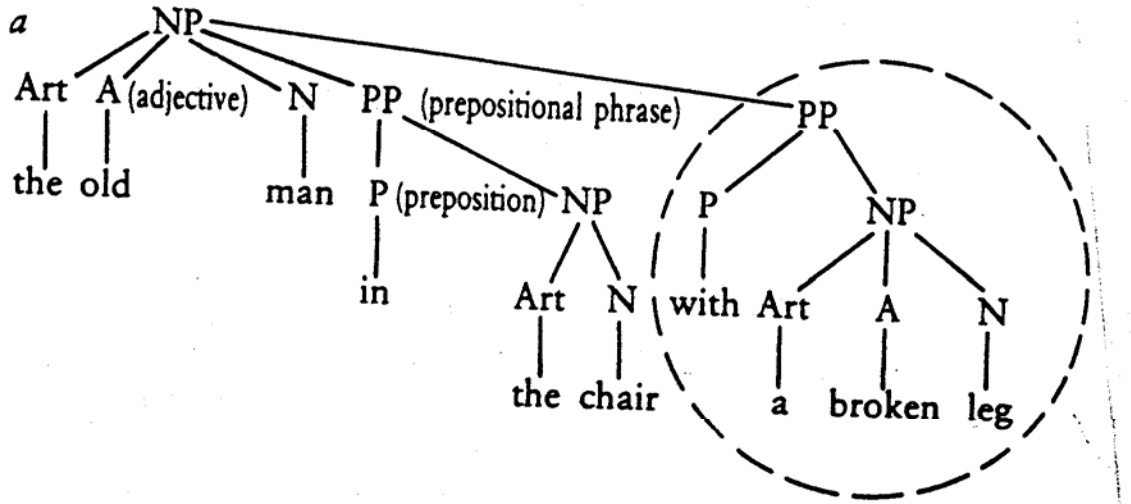
Art Adj N PP PP (high)

- OR

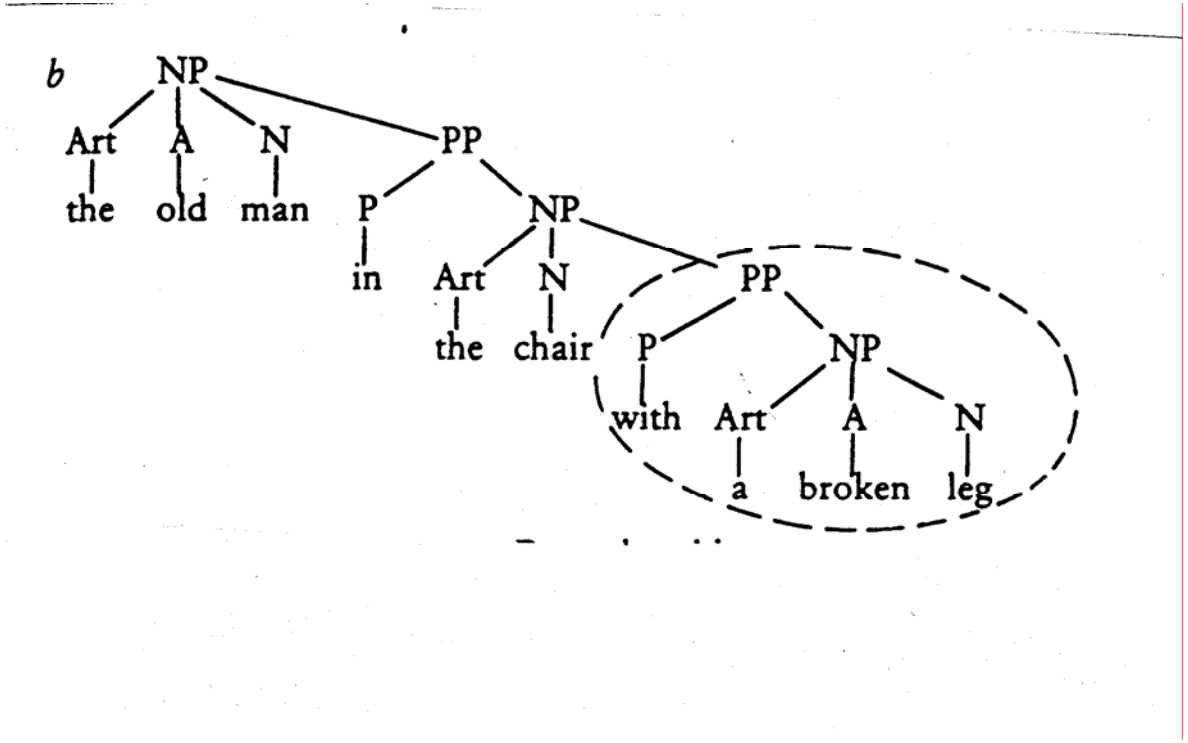
Art Adj N PP (low)

High Attachment (man's leg)

(14) the old man in the chair with a broken leg



Low Attachment (chair's leg)



Why speech perception is hard

- Rapid Rate
  - 15 phonemes/second
    - 67 ms/phoneme
  - 50 phonemes/second
    - 20 ms/phoneme
- Absence of Clear Boundaries
  - No “white space” as sounds blend into one another
  - Silence only for stop consonants and pauses
  - Parallel transmission or co-articulation
- Variability
  - Across speakers
  - Across registers
    - Yelled/Whispered/Sung
  - Across words
    - delight
    - dapper
    - dubious
- Low Quality of Information
  - 50% of words in normal speech unintelligible when presented in isolation

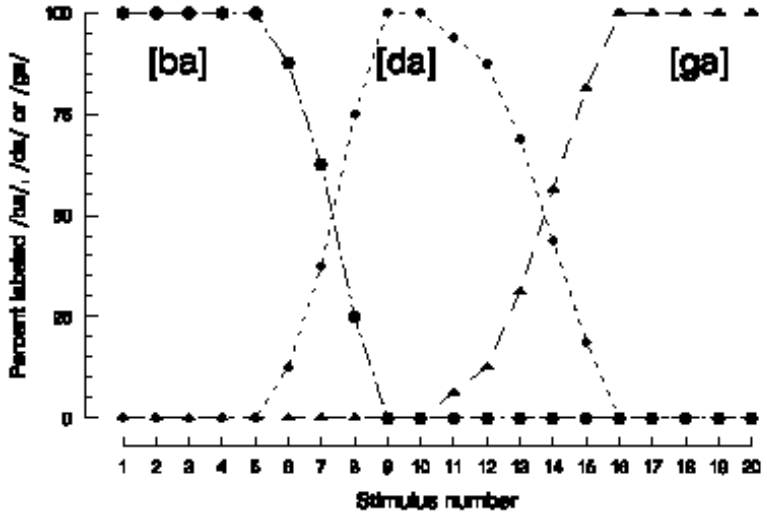
Theories of Speech Perception

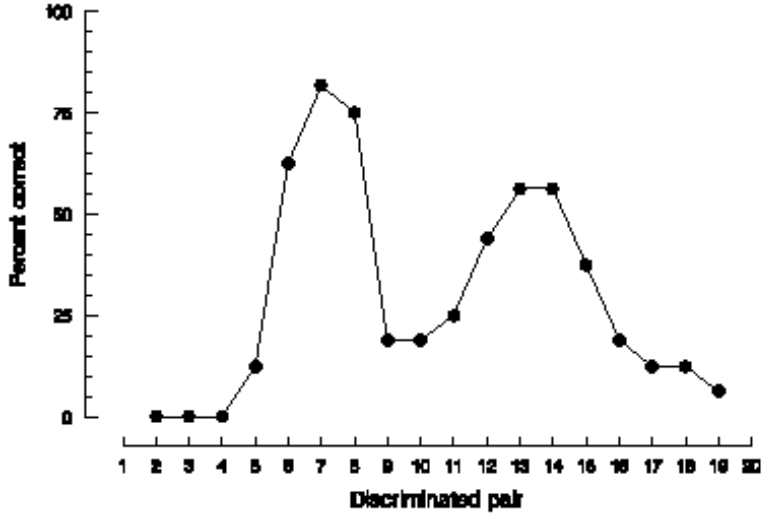
- Motor Theory (Liberman)
  - Close link between perception and production of speech
    - Use motor information to compensate for lack of invariants in speech signal
    - Determine which articulatory gesture was made, infer phoneme
  - Human speech perception is an innate, species-specific skill
    - Because only humans can produce speech, only humans can perceive it as a sequence of phonemes
    - Speech is special
- Auditory Theory
  - Derives from general properties of the auditory system
  - Speech perception is not species-specific

Empirical Evidence

- Knowledge of Articulatory Constraints seems to guide speech perception
  - Acoustic characteristics not constant across phones
  - [ba] confused with [da] but not with [sa]
  - Rated similarity between phonemes depends on number of shared articulatory features

What phenomenon is illustrated here?





Why was this viewed as evidence for the motor theory?

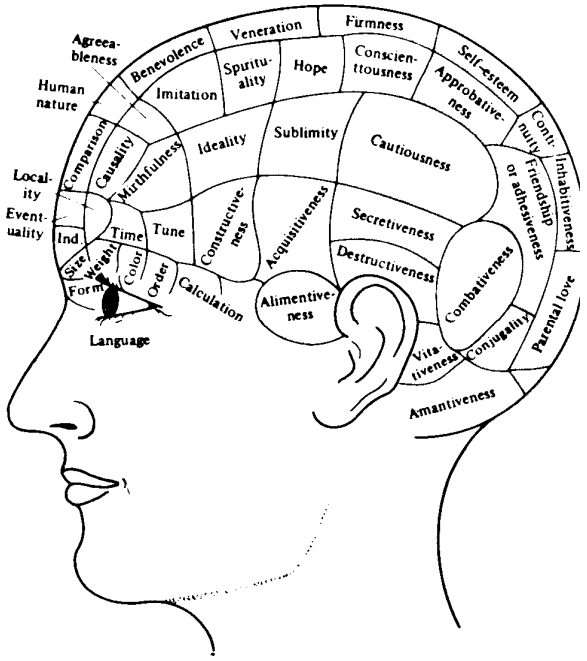
More questions

- What makes speech perception hard?
- What claims does the motor theory of speech perception make?
- What evidence supports the motor theory of speech perception?
- What evidence goes against the motor theory of speech perception?

Background

- Localization
  - Certain regions devoted to specific tasks.
  - Broca, Gall
  - Based on (almost) no evidence
- Equipotentiality
  - Whole brain involved with tasks
  - Flourens (1840s), Head, Lashley

- Both correct/wrong.



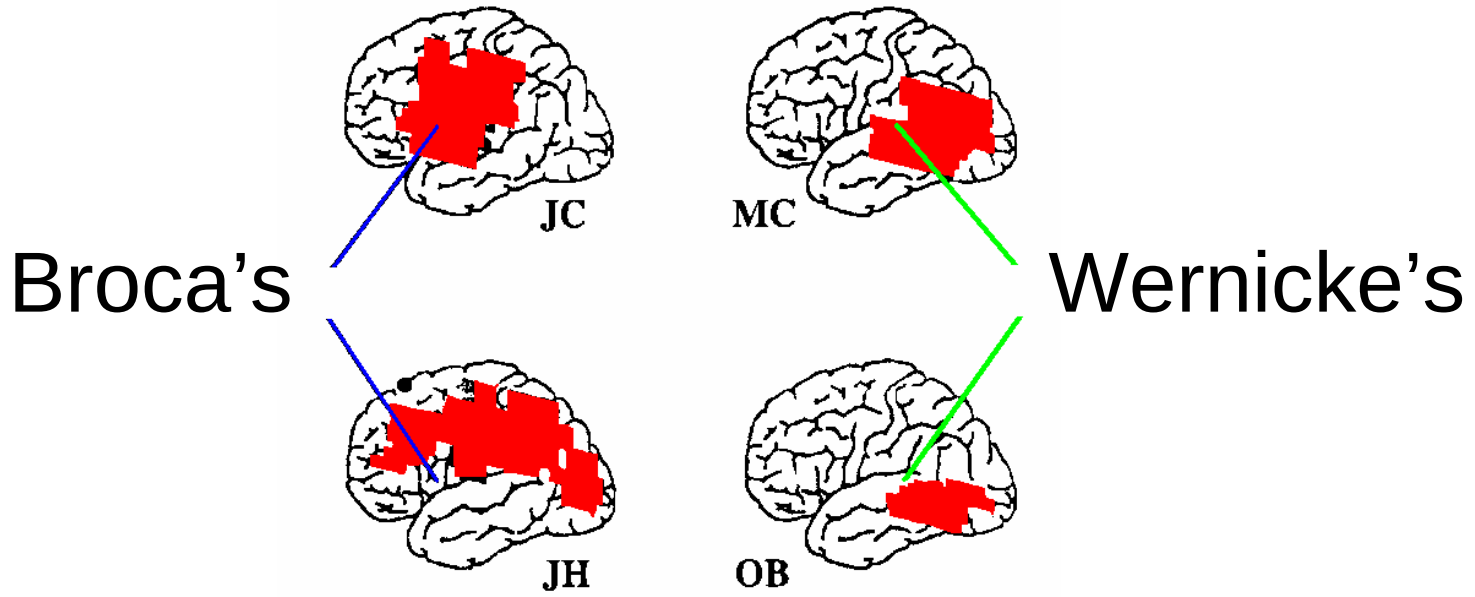
Classical Types of Aphasia

(- = relatively deficient; + = *relatively* spared)

APHASIA TYPE	NAMING	FLUENCY	COMPRE-HENSION	REPETITION
Broca's	-	-	+	-
Transcortical Motor	-	-	+	+
Wernicke's	-	+	-	-
Transcortical Sensory	-	+	-	+
Conduction	-	+	+	-
Anomia	-	+	+	+
Global	-	-	-	-

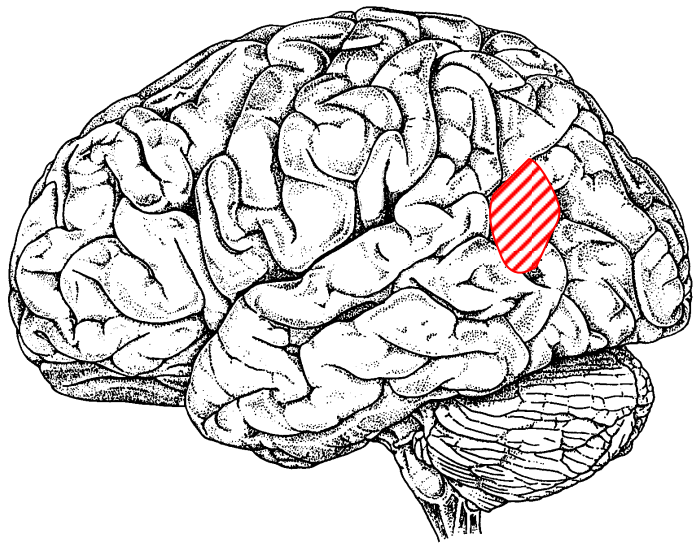
Exceptions to the rule

- Dronkers et al [2000] note exceptions to anatomy
- Traditional Theory:
  - JC will have Broca's and MC will have Wernicke's Aphasia
  - JH and OB will not have aphasia
- Reality:
  - Neither JC or MC has Aphasia
  - JH has Broca's and OB has Wernicke's Aphasia.



Alexia with agraphia

- Déjerine, 1891: Damage to the angular gyrus (BA 39) leads to
  - 'Alexia with agraphia' reading & writing deficits
  - Intact speech comprehension



Alexia without agraphia

- Disconnection of angular gyrus from visual inputs
  - Language outputs intact
  - Patients cannot read
  - Writing preserved

Rare: left and right pathways to angular gyrus

Requires damage to

1. posterior callosum
2. left occipital lobe

Without damage to left angular gyrus