Cognitive Modeling

Lecture 6: Models of Spoken Word Recognition

Sharon Goldwater

School of Informatics
University of Edinburgh
sgwater@inf.ed.ac.uk

January 28, 2010

Reading: Marslen-Wilson (1987).

Spoken Word Recognition

- All cognitively normal humans use language, need not be explicitly taught.
- For a familiar language, we perceive continuous speech stream as a sequence of discrete *words*.
    - Each word is an arbitrary correspondence between sound and meaning.
    - *Recognition*: identifying familiar sound sequence and its associated meaning.
- How do we recognize words, either in context or in isolation?

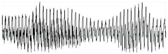
Illustration

Recognition may seem trivial – is there even a problem to study?

- Same information, different (visual) representation: no longer recognizable.

- Different instances of words can be very different due to speaker, pronunciation, context.





Marslen-Wilson (1987)

A classic paper on modeling spoken word recognition, example of a good modeling paper.

- Reviews many of the important issues in spoken word recognition.
- Presents a simple model (Cohort model) addressing several of these issues.
- Compares to previous models (notably, Logogen model).
- Lists several predictions of the Cohort model and how they were tested.
- Discusses weaknesses of the model and possible future extensions to address them.

Simplifying assumption

We abstract away from the continuous nature of the acoustic signal and use a symbolic input representation.

- Original Cohort model is based on *phonemes*: smallest units of sound that distinguish between words. (**b**ig vs. **d**ig).
  Input: l ʊ k æ t θ ə j ɛ l o d ɔ g
- For readability, I will use ordinary English characters and spelling.
  Input: `l o o k a t t h e y e l l o w d o g`

Psychological findings

Word recognition is *incremental* (online): humans need not hear the full word before recognition occurs.

- **Gating task** (listen to increasingly long word prefixes): recognition occurs when the prefix heard uniquely identifies the word (e.g. tres**p**ass, ora**n**ge). Marslen-Wilson (1987) calls this the *recognition point*. [1 2 3 4 5 6 7 8 9 10]
- **Lexical decision task** (words vs. non-words): reaction time for non-words is approximately constant from first non-word phoneme (e.g. tres**k**, oran**so**)
- **Phoneme monitoring task** (listen for a particular sound): reaction time is approximately constant from occurrence of phoneme or recognition point of word, whichever comes first.

Psychological findings

Word recognition is influenced by *context*: words can be recognized sooner in context than in isolation.

- phoneme monitoring and gating tasks show earlier recognition for words in sentence contexts:

  I eat fish but don't enjoy chi-
  Did you give the toys to the chi-

- Marslen-Wilson (1987) refers to this as *early selection*.

How do bottom-up (acoustic) and top-down (contextual) information interact during the recognition process?

Logogen model (Morton 1969)

Early model assumes each word is associated with a *logogen*: a unit with phonetic, syntactic, and semantic information. Logogens can be activated by perceptual or contextual factors.

- As more input is heard, activation rises for logogens whose phonetic representation matches the input.
- Activation also rises for logogens that match the current context.
- The first logogen to reach a certain threshold of activation is recognized.

Cohort model (Marslen-Wilson 1987)

Cohort model assumes initial activation of words is bottom-up. Active words are then *filtered* by context and later input.

- Activation from phonetic input: all words with the same initial phoneme are activated upon hearing the first phoneme. This is the *word-initial cohort*.
- Phonetic filtering: As more input is heard, some words in the cohort become incompatible with the input and are filtered out.
- Contextual filtering: words that are incompatible with the syntactic or semantic context are also filtered out.

Both activation and filtering are *parallel* processes that do not depend on the size of the cohort.

Example

Without context:

Heard:	o	or	ora	oran
Active:	often	oracle	oracle	orange
	oracle	orange	orange	
	orange	orb		
	orb	order		
	order	. . .		
	. . .			

Example

With context:

Heard: "The room is painted a hideous shade of..."

	o	or
Active:	often	orange
	oracle	
	orange	
	orb	
	order	
	...	

(Note that timing of context filtering is vague. Perhaps it is not as fast as shown here.)

Cohort vs. Logogen

Marslen-Wilson (1987) discusses several advantages of Cohort. Two main ones are

- Non-word identification: Because Logogen has only positive activation, it must wait until the end of the input to identify a non-word.
- Recognition points: In Logogen, recognition of a word doesn't depend on whether other words are possible or not. A word might not reach activation threshold until well after the point at which no other words are possible.

Cogent model: Experimental environment

We will consider a model for recognizing isolated words only.
Experimental environment contains

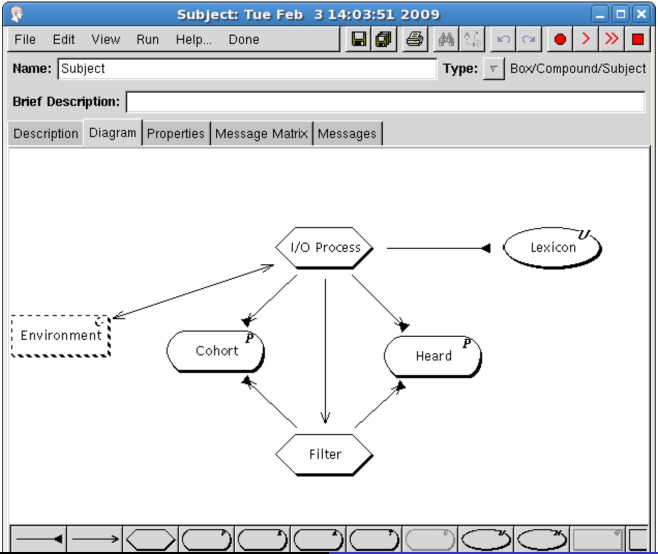
- **Stimuli**: the words to be recognized, represented as lists of phonemes ending with '.' to indicate silence at end of word.

  Example: `stimulus([b, i, g, .])`

- **Experimenter**. Contains one rule, which waits until previous word has been recognized, then sends the next word:

  TRIGGER: `system_quiescent`
  IF: `stimulus(Phonemes)` is in **Stimuli**
  THEN: send `recognize(Phonemes)` to **I/O Process**

Cogent model: Subject



Cogent model: Subject

Basic idea:

- Get all words from **Lexicon** that match first input phoneme, add them to **Cohort**, and start the filtering process.
- Filtering is recursive: examine the current input phoneme, remove words from **Cohort** whose next phoneme doesn't match, then move on to the next input phoneme and reduce the to-be-matched part of the **Cohort** words by one phoneme.
- While filtering, keep track of which phonemes have been heard and filtered already in **Heard**.
- When only one word remains, output the word and the contents of **Heard** to indicate the recognition point.

List syntax

- List consists of comma-separated terms enclosed in square brackets. Ex: `[a,b,c]`, `[X]`, `[]`;
- The '|' symbol is used to separate the *head* and *tail* of a list. Ex: `[a,b|Rest]`, `[X|Y]`;
  - Head: one or more terms.
  - Tail: a single variable representing the remainder of the list. The tail is a list also, i.e. will only unify with other lists.
- Special variable '_' can be used as a "don't care" when it's unnecessary to reuse the value. Think of each instance of '_' as a uniquely named variable.

Terms	Unifies as	Bindings
`[a,b,c],[X|Y]`	`[a,b,c]`	`X` → `a`, `Y` → `[b,c]`
`[a,b],[X1,X2|Y]`	`[a,b,c]`	`X1` → `a`, `X2` → `b`, `Y` → `[]`
`[a,b,c],[X|_]`	`[a,b,c]`	`X` → `a`
`[], [X|_]`	fails	

Lexicon contents

We use a toy lexcion:

cat	house
category	bread
catch	big
dog	bag
horse	bid

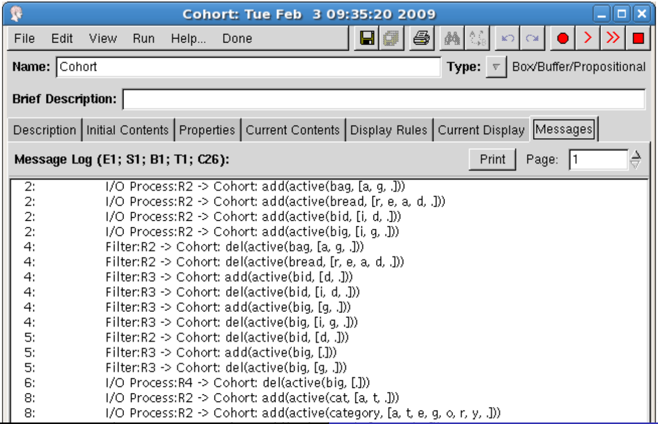
Words represented as strings of phonemes (characters):

- Ex: `word(cat,[c,a,t])`

Cohort messages

Cohort keeps track of active words and remainder to be matched.

Recognizing the word *big*:



I/O Process

Create the word-initial cohort and starts the filtering process:

Rule 1 (unrefracted): *Remember that we've heard the first phoneme and start filtering*
TRIGGER: recognize([P|Ps])
IF: True
THEN: delete heard(_) from Heard
add heard([P]) to Heard
send start_filtering(Ps) to I/O Process

Rule 2 (unrefracted): *Add words from lexicon to cohort that match the first phoneme we heard*
TRIGGER: recognize([P|_])
IF: word(Word, [P|Ps]) is in Lexicon
THEN: add active(Word, Ps) to Cohort

I/O Process

Monitor the cohort and output result when only one word left:

Rule 3 (unrefracted): *If cohort contains multiple words, start filtering on remaining input*
TRIGGER: start_filtering(RestPs)
IF: exists active(Word, _) is in Cohort
not unique active(Word, _) is in Cohort
THEN: send filter(RestPs) to Filter

Rule 4 (unrefracted): *If only one word in cohort, recognize it and reinitialize*
IF: unique active(Word, Xs) is in Cohort
heard(Heard) is in Heard
THEN: send recognized(Word, Heard) to Environment:Output
delete active(Word, Xs) from Cohort
delete heard(Heard) from Heard
add heard([]) to Heard

Filter

Match the current phoneme to words in the cohort, removing any that don't match.

Rule 1 (unrefracted): *Append current phoneme to prefix in Heard*
TRIGGER: filter([P|_])
IF: heard(Beginning) is in Heard
NewBeginning results from appending Beginning to [P]
THEN: delete heard(Beginning) from Heard
add heard(NewBeginning) to Heard

Rule 2 (unrefracted): *Delete words from cohort whose next phoneme doesn't match next input phoneme*
TRIGGER: filter([P|Ps])
IF: active(Word, [W|Ws]) is in Cohort
P is distinct from W
THEN: delete active(Word, [W|Ws]) from Cohort

Rule 3 (refracted): *If no more input, delete all words with at least one remaining phoneme*
TRIGGER: filter([.])
IF: active(Word, [P|_]) is in Cohort
P is distinct from .
THEN: delete all active(Word, _) from Cohort

Filter

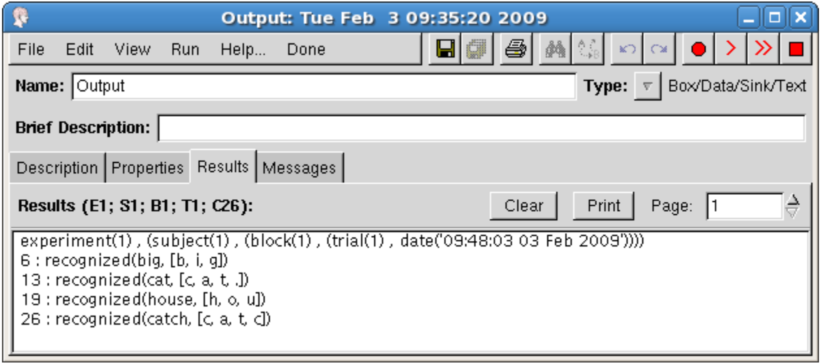
If more than one word is left, move to the next phoneme and recurse:

Rule 4 (refracted): *If there is a word with matching first phoneme, remove phoneme to prepare for recursion*
TRIGGER: filter([P|Ps])
IF: active(Word, [P|Ws]) is in Cohort
THEN: delete active(Word, [P|Ws]) from Cohort
add active(Word, Ws) to Cohort

Rule 5 (unrefracted): *If multiple matching words left, recurse*
TRIGGER: filter([P|RestPs])
IF: exists active(_, [P|_]) is in Cohort
not unique active(_, [P|_]) is in Cohort
THEN: send filter(RestPs) to Filter

Output

Output when recognizing *big*, *cat*, *house*, *catch*:



Additional predictions of Cohort

Parallel processing in activation and filtering predicts that the size of the cohort should not affect the speed of recognition.

- *Supported by evidence from lexical decision*: Response time to non-word does not depend on the number of words in the "terminal cohort".

Bottom-up activation predicts that even contextually inappropriate words will be briefly activated.

- *Supported by evidence from cross-modal priming*: targets that are semantically related to words in the cohort (including contextually inappropriate words) are primed if visual lexical decision is presented before recognition point of auditory stimulus.

Problems with Cohort

Cohort model fails to account for several aspects of recognition:

- Frequency effects: After controlling for recognition point, more frequent words are recognized faster than less frequent words. Cohort predicts no effect.
- Contextually anomalous words: Cohort predicts that these cannot be recognized.

  "The room is painted a hideous shade of oracle"
- Mispronunciations/misperceptions: Cohort cannot overcome these, since correct word will be knocked out of the cohort or never enter it.

Marslen-Wilson (1987) suggests some ways to address these issues. How would you do it?

Summary

- Key features of the Cohort model: parallel processing, bottom-up activation, top-down filtering.
- Model accounts for recognition points of isolated words, reject points of non-words, and early selection of words in context.
- Lack of robustness due to symbolic input representation and activation levels.

References

Marslen-Wilson, W. 1987. Functional parallelism in spoken word-recognition. *Cognition* 25:71–102.

Morton, J. 1969. Interaction of information in word recognition. *Psychological Review* 76:165–178.