

Natural Language Processing

CSE 592 Applications of AI
Winter 2003

Speech Recognition
Parsing
Semantic Interpretation

1

NLP Research Areas

- Speech recognition: convert an acoustic signal to a string of words
- Parsing (syntactic interpretation): create a parse tree of a sentence
- Semantic interpretation: translate a sentence into the representation language.
 - Disambiguation: there may be several interpretations. Choose the most probable
 - Pragmatic interpretation: incorporate current situation into account.

2

Some Difficult Examples

- From the newspapers:
 - Squad helps dog bite victim.
 - Helicopter powered by human flies.
 - Levy won't hurt the poor.
 - Once-sagging cloth diaper industry saved by full dumps.
- Ambiguities:
 - Lexical: meanings of 'hot', 'back'.
 - Syntactic: I heard the music in my room.
 - Referential: The cat ate the mouse. It was ugly.

3

Overview

- Speech Recognition:
 - Markov model over small units of sound
 - Find most likely sequence through model

4

Overview

- Speech Recognition:
 - Markov model over small units of sound
 - Find most likely sequence through model
- Parsing:
 - Context-free grammars, plus agreement of syntactic features

5

Overview

- Speech Recognition:
 - Markov model over small units of sound
 - Find most likely sequence through model
- Parsing:
 - Context-free grammars, plus agreement of syntactic features
- Semantic Interpretation:
 - Disambiguation: word tagging (using Markov models again!)
 - Logical form: unification

6

Speech Recognition

- Human languages are limited to a set of about **40 to 50 distinct sounds called phones: e.g.,**
 - [ey] bet
 - [ah] but
 - [oy] boy
 - [em] bottom
 - [en] button
- These phones are characterized in terms of **acoustic features, e.g., frequency and amplitude, that can be extracted from the sound waves**

7

Difficulties

- **Why isn't this easy?**
 - just develop a dictionary of pronunciation
e.g., coat = [k] + [ow] + [t] = [kowt]
 - but: recognize speech ≈ wreck a nice beach
- **Problems:**
 - homophones: different fragments sound the same
 - e.g., rec and wreck
 - segmentation: determining breaks between words
 - e.g., nize speech and nice beach
 - signal processing problems

8

Speech Recognition Architecture

• Large vocabulary, continuous speech (words not separated), speaker-independent

9

Signal Processing

- **Sound is an analog energy source resulting from pressure waves striking an eardrum or microphone**
- **A device called an analog-to-digital converter can be used to record the speech sounds**
 - sampling rate: the number of times per second that the sound level is measured
 - quantization factor: the maximum number of bits of precision for the sound level measurements
- e.g., telephone: 3 KHz (3000 times per second)
 - e.g., speech recognizer: 8 KHz with 8 bit samples so that 1 minute takes about 500K bytes

10

Signal Processing

- **Wave encoding:**
 - group into ~10 msec frames (larger blocks) that are analyzed individually
 - frames overlap to ensure important acoustical events at frame boundaries aren't lost
 - frames are analyzed in terms of features, e.g.,
 - amount of energy at various frequencies
 - total energy in a frame
 - differences from prior frame
 - vector quantization further encodes by mapping frame into regions in n-dimensional feature space

11

Signal Processing

- **Goal is speaker independence so that representation of sound is independent of a speaker's specific pitch, volume, speed, etc. and other aspects such as dialect**
- **Speaker identification does the opposite, i.e. the specific details are needed to decide who is speaking**
- **A significant problem is dealing with background noises that are often other speakers**

12

Speech Recognition Model

- Bayes's Rule is used break up the problem into manageable parts:

$$P(\text{words}|\text{signal}) = \frac{P(\text{words})P(\text{signal} | \text{words})}{P(\text{signal})}$$
 - $P(\text{signal})$: is ignored (normalizing constant)
 - $P(\text{words})$: Language model
 - likelihood of words being heard
 - e.g. "recognize speech" more likely than "wreck a nice beach"
 - $P(\text{signal}|\text{words})$: Acoustic model
 - likelihood of a signal given words
 - accounts for differences in pronunciation of words
 - e.g. given "nice", likelihood that it is pronounced [nuys] etc.

13

Language Model (LM)

- $P(\text{words})$ is the joint probability that a sequence of words = $w_1 w_2 \dots w_n$ is likely for a specified natural language
- This joint probability can be expressed using the chain rule (order reversed):

$$P(w_1 w_2 \dots w_n) = P(w_1) P(w_2 | w_1) P(w_3 | w_1 w_2) \dots P(w_n | w_1 \dots w_{n-1})$$
- Collecting the probabilities is too complex; it requires statistics for m^{n-1} starting sequences for a sequence of n words in a language of m words
- Simplification is necessary

14

Language Model (LM)

- First-order Markov Assumption says the probability of a word depends only on the previous word:

$$P(w_i | w_1 \dots w_{i-1}) \approx P(w_i | w_{i-1})$$
- The LM simplifies to

$$P(w_1 w_2 \dots w_n) = P(w_1) P(w_2 | w_1) P(w_3 | w_2) \dots P(w_n | w_{n-1})$$
 - called the bigram model
 - it relates consecutive pairs of words

15

Language Model (LM)

- More context could be used, such as the two words before, called the trigram model, but it's difficult to collect sufficient data to get accurate probabilities
- A weighted sum of unigram, bigram, trigram models could be used as a good combination:

$$P(w_1 w_2 \dots w_n) = c_1 P(w_i) + c_2 P(w_i | w_{i-1}) + c_3 P(w_i | w_{i-1} w_{i-2})$$
- Bigram and trigram models account for:
 - local context-sensitive effects
 - e.g. "bag of tricks" vs. "bottle of tricks"
 - some local grammar
 - e.g. "we was" vs. "we were"

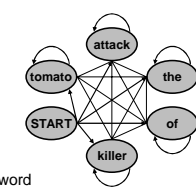
16

Language Model (LM)

- Probabilities are obtained by computing statistics of the frequency of all possible pairs of words in a large training set of word strings :
 - if "the" appears in training data 10,000 times and it's followed by "clock" 11 times then $P(\text{clock} | \text{the}) = 11/10000 = .0011$
- These probabilities are stored in:
 - a probability table
 - a probabilistic finite state machine
- Good-Turing estimator:
 - total mass of unseen events \approx total mass of events seen a single time

17

Language Model (LM)

- Probabilistic finite state machine: a (almost) fully connected directed graph:
 
- nodes (states): all possible words and a START state
- arcs: labeled with a probability
 - from START to a word is the prior probability of the destination word
 - from one word to another is the probability of the destination word given the source word

18

Language Model (LM)

- Probabilistic finite state machine: a (almost) fully connected directed graph:
- joint probability is estimated for *bigram* model by starting at START and multiplying the probabilities of the arcs that are traversed for a given sentence/phrase
- $P(\text{"attack of the killer tomato"}) = P(\text{attack})P(\text{of} | \text{attack})P(\text{the} | \text{of})P(\text{killer} | \text{the})P(\text{tomato} | \text{killer})$

19

Acoustic Model (AM)

- * $P(\text{signal} | \text{words})$ is the conditional probability that a signal is likely given a sequence of words for a particular natural language
- This is divided into two probabilities:
 - $P(\text{phones} / \text{word})$: probability of a sequence of phones given word
 - $P(\text{signal} / \text{phone})$: probability of a sequence of vector quantization values from the acoustic signal given phone

20

Acoustic Model (AM)

- $P(\text{phones} / \text{word})$ can be specified as a Markov model, which is a way of describing a process that goes through a series of states, e.g. tomato:
- nodes (states): corresponds to the production of a phone
 - sound slurring (co-articulation) typically from quickly pronouncing a word
 - variation in pronunciation of words typically due to dialects
- arcs: probability of transitioning from current state to another

21

Acoustic Model (AM)

- $P(\text{phones} / \text{word})$ can be specified as a Markov model, which is a way of describing a process that goes through a series of states, e.g., tomato:
- $P(\text{phones} / \text{word})$ is a path through the diagram, i.e.,
 - $P(\text{[towmeytow]} | \text{tomato}) = 0.2 * 1 * 0.5 * 1 * 1 = 0.1$
 - $P(\text{[towmaatow]} | \text{tomato}) = 0.2 * 1 * 0.5 * 1 * 1 = 0.1$
 - $P(\text{[tahmeytow]} | \text{tomato}) = 0.8 * 1 * 0.5 * 1 * 1 = 0.4$
 - $P(\text{[tahmaatow]} | \text{tomato}) = 0.8 * 1 * 0.5 * 1 * 1 = 0.4$

22

Acoustic Model (AM)

- $p(\text{signal} / \text{phone})$ can be specified as a hidden Markov model (HMM), e.g. [m]:

C1: 0.5	C3: 0.2	C4: 0.1
C2: 0.2	C4: 0.7	C6: 0.5
C3: 0.3	C5: 0.1	C7: 0.4
- nodes (states): probability distribution over a set of vector quantization values
- arcs: probability of transitioning from current state to another
- phone graph is technically a HMM since states aren't unique

23

Acoustic Model (AM)

- $P(\text{signal} / \text{phone})$ can be specified as a hidden Markov model (HMM), e.g., [m]:

C1: 0.5	C3: 0.2	C4: 0.1
C2: 0.2	C4: 0.7	C6: 0.5
C3: 0.3	C5: 0.1	C7: 0.4
- $P(\text{signal} / \text{phone})$ is a path through the diagram, i.e.,
 - $P(\text{[C1,C4,C6]} | \text{[m]}) = (0.7 * 0.1 * 0.6) * (0.5 * 0.7 * 0.5) = 0.00735$
 - $P(\text{[C1,C4,C4,C6]} | \text{[m]}) = (0.7 * 0.9 * 0.1 * 0.6) * (0.5 * 0.7 * 0.7 * 0.5) + (0.7 * 0.1 * 0.4 * 0.6) * (0.5 * 0.7 * 0.1 * 0.5) = 0.0049245$
- * This allows for variation in speed of pronunciation

24

Combining Models

tomato

[m]

Onset Mid End FINAL

C1: 0.5 C2: 0.2 C3: 0.3 C4: 0.7 C5: 0.1 C6: 0.5 C7: 0.4

Create one large HMM

25

Viterbi Algorithm

```

function VITERBI(observations of len T, state-graph) returns best-path
    num-states ← NUM-OF-STATES(state-graph)
    Create a path probability matrix viterbi[num-states+2, T+2]
    viterbi[0, 0] ← 1.0
    for each time step t from 0 to T do
        for each state s from 0 to num-states do
            for each transition s' from s specified by state-graph
                new-score ← viterbi[s, t] * a[s, s'] * bs'(ot)
                if ((viterbi[s', t+1] = 0) || (new-score > viterbi[s', t+1]))
                    then
                        viterbi[s', t+1] ← new-score
                        back-pointer[s', t+1] ← s
    Backtrace from highest probability state in the final column of viterbi[] and
    return path
    
```

26

Summary

- **Speech recognition systems work best if**
 - good signal (low noise and background sounds)
 - small vocabulary
 - good language model
 - pauses between words
 - trained to a specific speaker
- **Current systems**
 - vocabulary of ~200,000 words for single speaker
 - vocabulary of <2,000 words for multiple speakers
 - accuracy in the high 90%

27

Break

28

Parsing

29

Parsing

- Context-free grammars:
 - EXPR -> NUMBER
 - EXPR -> VARIABLE
 - EXPR -> (EXPR + EXPR)
 - EXPR -> (EXPR * EXPR)
- (2 + X) * (17 + Y) is in the grammar.
- (2 + (X)) is not.
- Why do we call them context-free?

30

Using CFG's for Parsing

- Can natural language syntax be captured using a context-free grammar?
 - Yes, no, sort of, for the most part, maybe.
- Words:
 - nouns, adjectives, verbs, adverbs.
 - Determiners: the, a, this, that
 - Quantifiers: all, some, none
 - Prepositions: in, onto, by, through
 - Connectives: and, or, but, while.
 - Words combine together into phrases: NP, VP

31

An Example Grammar

- $S \rightarrow NP VP$
- $VP \rightarrow V NP$
- $NP \rightarrow NAME$
- $NP \rightarrow ART N$
- $ART \rightarrow a | the$
- $V \rightarrow ate | saw$
- $N \rightarrow cat | mouse$
- $NAME \rightarrow Sue | Tom$

32

Example Parse

- *The mouse saw Sue.*

33

Ambiguity

- $S \rightarrow NP VP$ “Sue bought the cat biscuits”
- $VP \rightarrow V NP$
- $VP \rightarrow V NP NP$
- $NP \rightarrow N$
- $NP \rightarrow N N$
- $NP \rightarrow Det NP$
- $Det \rightarrow the$
- $V \rightarrow ate | saw | bought$
- $N \rightarrow cat | mouse | biscuits | Sue | Tom$

34

Chart Parsing

- Efficient data structure & algorithm for CFG's – $O(n^3)$
- Compactly represents all possible parses
 - Even if there are exponentially many!
- Combines top-down & bottom-up approach
 - Top down: what categories could appear next?
 - Bottom up: how can constituents be combined to create a instance of that category?

35

Augmented CFG's

- Consider:
 - Students like coffee.
 - Todd likes coffee.
 - Todd like coffee.

36

Augmented CFG's

- Consider:
 - Students like coffee.
 - Todd likes coffee.
 - Todd like coffee.
- $S \rightarrow NP[\text{number}] VP[\text{number}]$
 $NP[\text{number}] \rightarrow N[\text{number}]$
 $N[\text{number}=\text{singular}] \rightarrow \text{"Todd"}$
 $N[\text{number}=\text{plural}] \rightarrow \text{"students"}$
 $VP[\text{number}] \rightarrow V[\text{number}] NP$
 $V[\text{number}=\text{singular}] \rightarrow \text{"likes"}$
 $V[\text{number}=\text{plural}] \rightarrow \text{"like"}$

37

Augmented CFG's

- Consider:
 - I gave hit John.
 - I gave John the book.
 - I hit John the book.
- What kind of feature(s) would be useful?

38

Semantic Interpretation

- Our goal: to translate sentences into a logical form.
- But: sentences convey more than true/false:
 - It will rain in Seattle tomorrow.
 - Will it rain in Seattle tomorrow?
- A sentence can be analyzed by:
 - propositional content, and
 - speech act: tell, ask, request, deny, suggest

39

Propositional Content

- Target language: precise & unambiguous
 - Logic: first-order logic, higher-order logic, SQL, ...
- Proper names \rightarrow objects (Will, Henry)
- Nouns \rightarrow unary predicates (woman, house)
- Verbs \rightarrow
 - transitive: binary predicates (find, go)
 - intransitive: unary predicates (laugh, cry)
- Determiners most, some \rightarrow quantifiers

40

Semantic Interpretation by Augmented Grammars

- Bill sleeps.
- $S \rightarrow NP VP \{ VP.sem(NP.sem) \}$
 $VP \rightarrow \text{"sleep"} \{ \lambda x . sleep(x) \}$
 $NP \rightarrow \text{"Bill"} \{ BILL_962 \}$

41

Semantic Interpretation by Augmented Grammars

- Bill hits Henry.
- $S \rightarrow NP VP \{ VP.sem(NP.sem) \}$
 $VP \rightarrow V NP \{ V.sem(NP.sem) \}$
 $V \rightarrow \text{"hits"} \{ \lambda y,x . hits(x,y) \}$
 $NP \rightarrow \text{"Bill"} \{ BILL_962 \}$
 $NP \rightarrow \text{"Henry"} \{ HENRY_242 \}$

42

Montague Grammar

If your thesis is quite indefensible
Reach for semantics intensional.
Your committee will stammer
Over Montague grammar
Not admitting it's incomprehensible.

43

Coping with Ambiguity: Word Sense Disambiguation

- How to choose the best parse for an ambiguous sentence?
- If category (noun/verb/...) of every word were known in advance, would greatly reduce number of parses
 - Time flies like an arrow.
- Simple & robust approach: word tagging using a word bigram model & Viterbi algorithm
 - No real syntax!
 - Explains why "Time flies like a banana" sounds odd

44

Experiments

- Charniak and Colleagues did some experiments on a collection of documents called the "Brown Corpus", where tags are assigned by hand.
- 90% of the corpus are used for training and the other 10% for testing
- They show they can get 95% correctness with HMM's.
- A really simple algorithm: assign t to w by the highest probability tag $P(t|w) \rightarrow 91\%$ correctness!

45

Ambiguity Resolution

- Same approach works well for word-sense ambiguity
- Extend bigrams with 1-back bigrams:
 - John is blue.
 - The sky is blue.
- Can try to use other words in sentence as well – *e.g.* a naïve Bayes model
- Any reasonable approach gets about 85-90% of the data
 - Diminishing returns on "AI-complete" part of the problem

46

Natural Language Summary

- Parsing:
 - Context free grammars with features.
- Semantic interpretation:
 - Translate sentences into logic-like language
 - Use statistical knowledge for word tagging, can drastically reduce ambiguity – determine which parses are most likely
- Many other issues!
 - Pronouns
 - Discourse – focus and context

47