CS 545
Midterm Review

Announcement

- 4/12 General Review today
  ~ Practice midterm out today
- 4/17 Go over practice midterm
- 4/19 In class midterm

Midterm Format

- 2-3 Short answer (multiple choice?) questions, testing your knowledge of basic definitions.
- 2 Longer form questions
Midterm Topics

- Linguistics: Morphology / Syntax / Semantics / Discourse
- HMM's for part-of-speech. Viterbi algorithm
- PCFG's for parsing. CKY algorithm.

Different Levels of Linguistic Knowledge

- discourse
- pragmatics
- semantics
- syntax
- morphology
- phonology
- orthography
- phonetics

What are the sounds?
Different Levels of Linguistic Knowledge

- phonetics
- phonology
- morphology
- syntax
- semantics
- pragmatics
- discourse
- orthography

How can sounds combine?

What are the symbols?

第二阶段的奥运会体育比赛门票与残奥会开闭幕式门票的预订工作已经结束，现在进入门票分配阶段。在此期间，我们不再接受新的门票预订申请。

“Let’s go see the praade!”
“They have a mouse in they’re house.”
“What do you want for desert?”

What are the symbols?
Different Levels of Linguistic Knowledge

- discourse
- pragmatics
- semantics
- syntax
- morphology
- orthography

What are the words?

fax, google, w00t, OMG, Man-fucking-hattan, lol, lolz, unfriend, tweet, Obamacare, coo af

After it sorts each sub-part, it merges them.
After they sort each sub-part, they merge them.
How many merges are needed?
One merge.
Merging is fast.
To split is human, to merge divine.

morphology
What are the words?

One house among many houses
One mouse among many mice
uygarlaştıramadıklarımızdanmışınızcasin
"(behaving) as if you are among those whom we could not civilize"
Noah gave Kevin the book.
= Noah gave the book to Kevin.
= The book was given to Kevin by Noah.
= The book was given by Noah to Kevin.
*Gave Noah Kevin the book.

I want a flight to Tokyo.
I want to fly to Tokyo.
I found a flight to Tokyo.
*I found to fly to Tokyo.

**Different Levels of Linguistic Knowledge**

```
{}
  | discourse
  | |
  | pragmatics
  | |
  | semantics
  | |
  | syntax
  | |
  | morphology
  | |
  phonology
  | |
  | orthography
| phonetics
```

"Jerusalem - there is no such city!"

In this country a woman gives birth every fifteen minutes. Our job is to find that woman and stop her.

Colorless green ideas sleep furiously.
Different Levels of Linguistic Knowledge

- discourse
- pragmatics
- semantics
- syntax
- morphology
- phonology
- orthography
- phonetics

What are the intentions?

- pragmatics

I'm sorry Dave, I'm afraid I can't do that.
You're so funny.
I can't believe I ate the whole thing.

Different Levels of Linguistic Knowledge

- discourse
- pragmatics
- semantics
- syntax
- morphology
- phonology
- orthography
- phonetics

What's going on in context?
The Tin Woodman went to the Emerald City to see the Wizard of Oz and ask for a heart. After he asked for it, the Woodman waited for the Wizard’s response.

Any time you got nothing to do - and lots of time to do it - come on up.

Bernoulli and Multinomial

- coins and dice
- parameters
- data and likelihood
- maximum likelihood estimation of parameters

The Noisy-Channel Model
ASR System Components

Language Model

source \( P(w) \)

Acoustic Model

channel \( P(a|w) \)

decoder

best \( w \)

argmax \( P(w|a) = \arg\max_w P(a|w)P(w) \)

observed \( a \)

MT System Components

Language Model

source \( P(e) \)

Translation Model

channel \( P(f|e) \)

decoder

best \( e \)

argmax \( P(e|f) = \arg\max_e P(f|e)P(e) \)

observed \( f \)

Other Noisy-Channel Processes

- Spelling Correction
  \( P(\text{words} \mid \text{characters}) = P(\text{words})P(\text{characters} \mid \text{words}) \)
- Handwriting recognition
  \( P(\text{words} \mid \text{strokes}) = P(\text{words})P(\text{strokes} \mid \text{words}) \)
- OCR
  \( P(\text{words} \mid \text{pixels}) = P(\text{words})P(\text{pixels} \mid \text{words}) \)
- More…
The Unigram Model

- Every word gets some probability.
- We assume words are generated independently of each other.
- This means that:

\[ p(w | L) = \prod_{i=1}^{L} p(w_i) \]

\[ p(w) = \frac{1}{\sum_{L} \prod_{i=1}^{L} p(w_i)} \]

N-Gram Models

- Middle-ground between unigram and full history models.
- Markov assumption: a word only depends on a finite, most recent part of the history (N-1 words).

\[ p(w) = \left( \prod_{i=1}^{L} p(w_i | w_{i-n+1}^{i-1}) \right) \cdot p(\bullet | w_{L-n+1}^{i-1}) \]

Issues

- How many parameters?
- How to estimate them?
- Why is smoothing necessary?
- Add-delta smoothing
- What should the value of “n” be?
- How to interpolate different values of “n”
Smoothing

- Sparse data:
  \( P(w \mid \text{denied the}) \)
  3 allegations
  2 reports
  1 claims
  1 request
  7 total

- Smoothing flattens spiky distributions so they generalize better

\[
\hat{p}_{\text{La}}(w_1, ..., w_N) = \frac{1 + \text{count}(w_1 w_2 ... w_{N-1} w_N)}{\sum_{v \in \Sigma} \text{count}(w_1 w_2 ... w_{N-1} v)}
\]

\[
\hat{p}_{\text{MLE}}(w_N \mid w_1, ..., w_{N-1} \cdot) = \frac{\text{count}(w_1 w_2 ... w_{N-1} w_N)}{\sum_{v \in \Sigma} \text{count}(w_1 w_2 ... w_{N-1} v)}
\]

Smoothing Method: Laplace

- Add 1 (or delta) to every N-gram's count.
- Compare:

\[
\hat{p}_{\text{interp}}(w_N \mid w_1, ..., w_{N-1}) = \sum_{\lambda \in \Lambda} \lambda \hat{p}_{\text{MLE}}(w_N \mid w_1...w_{N-1})
\]

Smoothing Method: Linear Interpolation

- Key Idea: Use simpler models (smaller N) to help smooth more complex ones.
Frequentist (classical) Statistics

- Probability refers to limiting relative frequencies.
  - Probabilities are objective properties of the real world.
- Parameters are fixed, unknown constants. Because they are not fluctuating, no useful probability statements can be made about parameters.
- Instead we talk of estimators (e.g. ML) and confidence intervals.

\[
L(\theta) = \prod_i p(x_i; \theta) \\
LL(\theta) = \sum_i \log p(x_i; \theta)
\]

Maximum Likelihood estimator

\[
\theta_{ML} = \arg \max_\theta LL(\theta)
\]

Bayesian Statistics

- Probability describes degree of belief, not limiting frequency (did Albert Einstein drink a cup of tea on August 1, 1948?)
- We can make probability statements about parameters, even though they are fixed constants.
- We make inferences about a parameter \( \theta \) by producing a probability distribution over \( \theta \)

\[
p(\theta) \quad p(x|\theta) \quad p(\theta|x) = \frac{p(x|\theta)p(\theta)}{\int p(x|\theta)p(\theta)d\theta}
\]

\[
\theta_{MAP} = \arg \max_\theta \left\{ \log p(\theta) + \sum_i \log p(x_i|\theta) \right\}
\]

The Part-of-Speech Task

- **Input**: a tokenized sentence \( w = w_1 w_2 ... w_n \)
- **Output**: the POS tags for the sentence, \( t = t_1 t_2 ... t_n \)
Broad POS categories

- **open classes**
  - nouns
  - verbs
  - adjectives
  - adverbs

- **closed classes**
  - prepositions
  - determiners
  - pronouns
  - conjunctions
  - auxiliary verbs
  - particles
  - numerals

HMMs

- Two kinds of information:
  - What tag sequences are grammatical (likely)?
  - What tags make the words likely?

- **Bigram version:**
  \[
  p(t, w) = \left( \prod_{i=1}^{n} p(t_i | t_{i-1}) \times p(w_i | t_i) \right) p(\text{stop} | t_n)
  \]

- **Trigram version:**
  \[
  p(t, w) = \left( \prod_{i=1}^{n} p(t_i | t_{i-2}, t_{i-1}) \times p(w_i | t_i) \right) p(\text{stop} | t_n)
  \]

Issues

- How many parameters?
- How to choose “n”?
- Smoothing?
- How to estimate them using Maximum Likelihood?
- How to recover the most likely tag sequence for a sentence efficiently?
  - Viterbi. Runtime?
Viterbi Equations

V[0, q0] = 1

\[V[t, q_j] = \max_{q_i} V[t-1, q_i] \times a_{i,j} \times b_{j,w_t}\]

\[\text{goal} = \max_{q_i} V[n, q_i] \times a_{i,f}\]

Viterbi Pseudocode

- \[V[\cdot, \cdot] \leftarrow 0\]
- \[V[0, q_0] \leftarrow 1\]
- for \(t = 1 \ldots n\)
  - for \(j = 1 \ldots |Q|\)
    - for \(i = 1 \ldots |Q|\)
      - \[V[t, q_j] \leftarrow \max\{ V[t, q_j], V[t-1, q_i] \times a_{i,j} \times b_{j,w_t}\}\]
  - for \(i = 1 \ldots |Q|\)
    - \[V[n+1, q_f] \leftarrow \max\{ V[t+1, q_f], V[n, q_i] \times a_{i,f}\}\]
- return \(V[n+1, q_f]\)

Bill directed plays about English kings.
Viterbi

Bill directed plays about English kings.

\[ V[t, q_t] = \max_V V[t-1, q_t] \times a_{i,j} \times b_j \text{, } w_t \]

Parse Trees

The move followed a round of similar increases by other lenders, reflecting a continuing decline in that market

Phrase Structure Parsing

- Phrase structure parsing organizes syntax into constituents or brackets
- In general, this involves nested trees
- Lots of ambiguity
- Not the only kind of syntax

new art critics write reviews with computers
Constituency Tests

- How do we know what nodes go in the tree?

- Classic constituency tests:
  - Substitution by proform
  - Question answers
  - Movement
  - Conjunction

- Cross-linguistic arguments, too

Probabilistic (or Stochastic) CFG

- $G = (\Sigma, N, S, R)$
- $\Sigma$: Vocabulary of terminal symbols
- $N$: set of nonterminal symbols (AKA variables)
- $S \in N$: special start symbol
- $R$: Production rules of the form $X \rightarrow \alpha$
  where $X \in N$ (a nonterminal symbol) and $\alpha \in (N \cup \Sigma)^*$ (a sequence of terminals and nonterminals)
  - Each rule gets a probability.
  - Rules with the same left side sum to 1.
  - Each nonterminal has a "die."

Issues

- What kind of distributions?  How many parameters?
- How to estimate parameters using Maximum Likelihood?
- How to predict highest probability tree for a sentence efficient?
  - CKY algorithm
- Refining a grammar to make it more discriminative
### Probabilistic CKY

<table>
<thead>
<tr>
<th>span symbol</th>
<th>production</th>
<th>prob</th>
<th>Recursion</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C[i-1, i, x_i] ) = 1</td>
<td>( p(V \rightarrow x_i) )</td>
<td>( p(V \rightarrow YZ) \times C[j, j, Y] \times C[j, k, Z] )</td>
<td></td>
</tr>
<tr>
<td>( C[i-1, i] ) = ( p(V \rightarrow x_i) )</td>
<td>( C[V, x_i] )</td>
<td>( C[i, k, V] ) = ( \max_{j \in [i+1, ..., k-1]} { C[i, j, Y] \times C[j, k, Z] } )</td>
<td></td>
</tr>
<tr>
<td>( p(x_i</td>
<td>x_1, ..., x_n) ) = ( C[0, n, S] )</td>
<td>( C[x, i, k] ) = ( \max { C[X, i, k], p(X</td>
<td>X \rightarrow YZ) \times C[Y, i, j] \times C[Z, j, k] } )</td>
</tr>
</tbody>
</table>

### Probabilistic CKY Pseudocode

- **Input:** \( x, G \)
- **Output:** \( C \)
- for \( i = 1 \ldots n \)
  - \( C[A, i-1, i] = p(A \rightarrow x_i) \)
- for \( t = 2 \ldots n \)
  - for \( i = 0 \ldots n - t \)
    - \( k = i + t \)
    - for \( j = i + 1 \ldots k - 1 \)
      - \( C[X, i, k] = \max \{ C[X, i, k], p(X | X \rightarrow YZ) \times C[Y, i, j] \times C[Z, j, k] \} \)
  - return \( C[S, 0, n] \)

### Probabilistic CKY

- Exploits shared substructure to optimize over an exponentially large set of trees, in polynomial time.
- Similar to Viterbi algorithm, minimum edit distance, ...
Problems with PCFGs

- What's different between basic PCFG scores here?
- What (lexical) correlations need to be scored?

Grammar Refinement

- Lexicalization [Collins '99, Charniak '00]
- Structure Annotation [Johnson '98, Klein&Manning '03]