Midterm Preparation

Adam Meyers
New York University
Administrative Details

- Time: Thursday October 19, 2017
- Ask clarification questions during test
  - I especially want to fix errors such as typos
- Open Book, Open Notes
  - You can bring materials
  - Search the web
  - Do simple calculations with a calculator
  - **DO NOT:**
    - communicate with others (texting, email, phone)
    - write/run actual programs
- Put your name and ID number on all test materials
- I will take attendance: please bring your ID
The Purpose of the Midterm?

• Pedagogical purposes
  – Track whether students learned parts of curriculum and what may need further clarification
  – Provide a motivating force for students to study the “important” parts of the curriculum
  – Clarify how to prepare students to do final projects

• Administrative purpose: determine 1/4 of grade

• Possible conflict
  – A difficult test makes pedagogical sense
  – An “acceptable” average grade may make administrative sense

• Current Strategy: Motivate test based on pedagogical objectives, but make it as open book as possible
  – You can bring materials, search the web, etc., but I will want you to solve the problems posed by the test
Outline

• Linguistic Resources & Descriptive Linguistics
  – Especially Corpus Annotation
• Rules used by Automated Procedures
  – Ones covered in Class
• Algorithms Discussed in Class
• How does Evaluation Work
• Sample Midterm
Annotation

• You should be able to write usable specifications
• You should be able to annotate based on specifications
• You should understand some of the mechanics
  – Character offsets
  – A Markup language
  – BIO tags
• You should understand the difference between training, development and test corpora
Descriptive Linguistics

• The basic parts of speech and phrasal categories.
  – The difference between a determiners and an adjective
  – Verbs, prepositions, coordinate conjunctions
• How to manually divide sentences into tokens
• You should know how to identify the head of a phrase
• You should be able to draw a phrase structure tree
  modeling the linguistic analysis of a sentence
Common Difficulties with Phrase Structure

• SBAR → IN S
  – …. that this is a sentence.

• PP → IN NP
  – in the house, for freedom, on the clock, of chocolate, ...

• Coordinate conjunctions (and, or, but) link 2 constituents of the same type together
  – [NP [NP …] and [NP …]]
  – [VP [VP …] or [VP …]]
  – [S [S …] but [S ]]

• to infinitives are VPs, e.g. [VP to [VP go to the movies]]
Rules: Regular Expressions

• You should know how to write a basic regular expression
  – Decent coverage, but not over-generate too much

• You should know how to write a phrase structure rule including at least:
  – Context free rules
  – Left (or right) regular rules

• For a regular expression, you should be able to identify a set of phrase structure rules that describe the same language (set of strings)
Algorithms: Deterministic Finite State Machine

• Given:
  – Finite State Machine (FSM)
  – Input String

• Would the FSM recognize the string?

• Which sequence of states would be entered before recognition was complete?

• How would the FSM on the next slide process:
  – AababAB
  – AABB
DFSA for Regexp: $A(ab)^*ABB$?
Algorithms: Context-Free Generator

• Show steps for randomly generating a sentence given a lexicon and context-free grammar with start symbol $S$

• The start symbol is inserted into a stack.

• Repeat until stack is empty:
  – If the top most item is a terminal, pop it off and include it in the output.
  – Else, replace the top most item in the stack with the right hand of a rule of the form $X \rightarrow \text{Right Hand Side}$
Example of Generator

• Add S to top of empty stack
  – Stack is now: S

• Substitute NP VP for S
  – Stack is now: NP VP

• Substitute DT N PP for NP
  – Stack is now: DT N PP VP

• Substitute DT with the, pop off the (terminal)
  – Stack is now: N PP VP

• Etc.
Algorithms: The CKY parsing algorithm

• Fill in the triangular chart given a (short) sentence and a set of context free rules

• Remember
  – How the chart encodes start and end positions
  – That each rule is in Chomsky Normal Form
    • i.e., is binary branching

• See the next slide
**6\textsuperscript{th} Iteration of CKY Algorithm**

<table>
<thead>
<tr>
<th></th>
<th>The</th>
<th>clam</th>
<th>'s</th>
<th>group</th>
<th>had</th>
<th>knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>0</td>
<td>D [0,1]</td>
<td>NP [0,2]</td>
<td>POSSP [0,3]</td>
<td>NP [0,4]</td>
<td>S [0,5]</td>
<td>S [0,6]</td>
</tr>
<tr>
<td>1</td>
<td>N, NP [1,2]</td>
<td>POSSP [1,3]</td>
<td>NP [1,4]</td>
<td>S [1,5]</td>
<td>S [1,6]</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>POSS [2,3]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>V, VP [4,5]</td>
<td>VP [4,6]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>N, NP [5,6]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Language Models

• Calculate a probability for each word based on available information
• Bi-gram model (most info), backing off to unigram and then OOV
• Bigram and Unigram frequencies for 48 word corpus on next slide:
  – begin and end of sentence are tokens
  – Words occurring only once are counted as *oov*
• Corpus = 48 word Edward Lear Limerick, where each line is assumed to be a separate sentence

There was an Old Man of the North,
Who fell into a basin of broth;
But a laudable cook,
Fished him out with a hook,
Which saved that Old Man of the North.
Frequencies

• Unigram freqs:
  – 21: *oov*
  – 5: B_Sent, E_Sent
  – 3: of, a, *comma*
  – 2: old, man, the, north

• Bigram freqs:
  – 11: *oov* + *oov*
  – 5: B_Sent + *oov*
  – 3: *oov* + a, a + *oov*, *comma* + E_Sent, a + *oov*
  – 2: *oov* + old, *oov* + E_Sent, *oov* + *comma*, old + man, man + of, of + the, the + north
  – 1: *oov* + of, of + *oov*, north + *oov*, north + *comma*

• There are 48 tokens.

• What is the probability of the following sentence?
  – But a laudable cook, –> but + a + laudable + cook + ,
Probability of “but a laudable cook”

- **but** = \( \frac{\text{freq}(\text{B-sent} + \text{oov})}{\text{freq}(\text{B-sent})} = \frac{5}{5} = 1 \)
- **a** = \( \frac{\text{freq}(\text{oov} + a)}{\text{freq}(\text{oov})} = \frac{3}{21} \)
- **laudible** = \( \frac{\text{freq}(a + \text{oov})}{\text{freq}(a)} = \frac{3}{3} = 1 \)
- **cook** = \( \frac{\text{freq}(\text{oov} + \text{oov})}{\text{freq}(\text{oov})} = \frac{11}{21} \)
- **comma** = \( \frac{\text{freq}(\text{oov} + \text{comma})}{\text{freq}(\text{oov})} = \frac{2}{21} \)
- **E-sent** = \( \frac{\text{freq}(\text{comma} + \text{E-sent})}{\text{freq}(\text{comma})} = \frac{3}{3} = 1 \)

- Probability of **But a laudable cook**, is:
  \[ 1 \times \frac{3}{21} \times \frac{3}{3} \times \frac{11}{21} \times \frac{2}{21} \times 1 = 0.00713 \]
Viterbi Decoding of HMM for *rose pickles*

- **Likelihood:**
  - *rose*: NNP .01, NN .02, VBD .05
  - *pickles*: NNP .001, NNS .03, VBZ .05

- **Transition Probabilities:**
Rose Pickles

- **Likelihood:**
  - *rose*: NNP 0.01, NN 0.02, VBD 0.05
  - *pickles*: NNP 0.001, NNS 0.03, VBZ 0.05

- Fill in: max (previous X transition X likelihood)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1: Rose</th>
<th>2: Pickles</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NNP</td>
<td></td>
<td>.42 * 0.01</td>
<td><em>.3</em>.001</td>
<td></td>
</tr>
<tr>
<td>NNS</td>
<td></td>
<td></td>
<td><em>.0</em> .03</td>
<td></td>
</tr>
<tr>
<td>NN</td>
<td></td>
<td>.20 * 0.02</td>
<td><em>.24</em>.03</td>
<td></td>
</tr>
<tr>
<td>VBZ</td>
<td></td>
<td></td>
<td><em>.3</em>.05</td>
<td></td>
</tr>
<tr>
<td>VBD</td>
<td></td>
<td>.05 * 0.05</td>
<td>*.2</td>
<td></td>
</tr>
<tr>
<td>End</td>
<td></td>
<td></td>
<td>*.2</td>
<td>*.45</td>
</tr>
</tbody>
</table>

Likelihood:
- *rose*: NNP .01, NN .02, VBD .05
- *pickles*: NNP .001, NNS .03, VBZ .05

Fill in: max (previous X transition X likelihood)
**Rose/NNP Pickles/VBZ**

- **Likelihood:**
  - *rose*: NNP .01, NN .02, VBD .05
  - *pickles*: NNP .001, NNS .03, VBZ .05

- Fill in: max (previous X transition X likelihood)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1: Rose</th>
<th>2: Pickles</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Start</strong></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NNP</strong></td>
<td></td>
<td>.42 * .01</td>
<td><em>3</em>.001=1.26*10^-6</td>
<td></td>
</tr>
<tr>
<td><strong>NNS</strong></td>
<td></td>
<td></td>
<td><em>0</em>.03</td>
<td></td>
</tr>
<tr>
<td><strong>NN</strong></td>
<td>.20 * .02</td>
<td></td>
<td><em>5</em>.03=6*10^-5</td>
<td></td>
</tr>
<tr>
<td><strong>VBZ</strong></td>
<td></td>
<td></td>
<td><em>3</em>.05=6.3*10^-5</td>
<td></td>
</tr>
<tr>
<td><strong>VBD</strong></td>
<td>.05 * .05</td>
<td></td>
<td><em>1</em>.05</td>
<td></td>
</tr>
<tr>
<td><strong>End</strong></td>
<td></td>
<td></td>
<td><em>0</em>.05</td>
<td></td>
</tr>
</tbody>
</table>

- **Likelihood:**
  - *rose*: NNP .01, NN .02, VBD .05
  - *pickles*: NNP .001, NNS .03, VBZ .05

- Fill in: max (previous X transition X likelihood)
Common Evaluation Metrics

• If all instances are classified
  – Accuracy = Correct/All-Instances

• If only some instances are classified
  – Precision = Correct/Instances in System Output
  – Recall = Correct/Instances in Answer Key
  – F-measure = Mean of Precision and Recall
    • Harmonic Mean of Precision and Recall
      – \[ \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} \]
Sample Precision and Recall

• System for finding holiday names
• Exactly 10 correct holiday names in hand-coded corpus (the answer key)
• The system marks 12 holiday names, 8 of which match the ones in the answer key.
  – Precision = 8/12 = .67
  – Recall = 8/10 = .80
  – F-measure = 2/(1/.80+1/.67) = .73
TFIDF

• TFIDF – Property of Term with respect to a document
  – keyword suitability, representativeness of a topic, etc.
  – Uses: Doc Retrieval, Term Extraction, etc.
• TF = frequency in a document
• IDF = number of documents in sample divided by number of documents containing word
• TFIDF = TF * log(IDF)
• Example: “rock” occurs 10 times in document X. It occurs in 100 out of 3000 documents in collection. TFIDF = 10*log(3000/100) = 34.01
• *Use natural logarithms just to be standard (easier to grade)
  – Systems get same results (e.g., same ranking) using any base
Cosine Similarity Between Query and Document

\[ \text{Similarity}(A, B) = \frac{\sum_i a_i \times b_i}{\sqrt{\sum_i a_i^2 \times \sum_i b_i^2}} \]

Example:
- the terms in the vectors include: animal, vegetable, mineral, monkey, golf enthusiast
- The vector for the query is: [0,0,0,34,.8]
- The vector for a given document is: [1,2,3,4,5]
- What is the similarity?
Practice Midterms

• Sample midterm & answers online
  – More time-consuming than actual midterm. 10 questions on the sample vs 7 on actual midterm.

• Last Term's Midterm & answers
  – Should be comparable to this term's midterm.
General Test-taking Advice

• Test is a game –
  – not worth getting tense about
  – Staying calm makes it easier to think clearly

• Time may be an issue
  – Finish as many questions as possible
  – Budget time
    • 75 minutes/7 questions ≈ 10 minutes/question

• Show your work
  – It makes it easier justify partial credit