Intro to Computational Linguistics: Final Review Lecture

Adam Meyers
New York University
Summary

• Administrative Details
• Regular Expressions
  – Writing them applying them to data
• Phrase structure and POS
  – Annotation
  – Algorithms using them or for detecting them: Viterbi, HMM, CKY parsing
• TFIDF and Cosine similarity
• Sequence Labeling with BIO tags: Noun Groups and NEs
• Reference Resolution
  – Binding Theory, Hobbs Search
• Calculating Evaluation Measures
• Feature Structure (quick example)
• Machine Translation
  – EM and Decoding
• Following Annotation Guidelines/Doing Linguistic Analyses
• Additional Questions
Final Exam

• Open book, Open notes, calculator is OK
  – No email, texting, using programs for algorithms

• You have approximately 1 hour and 50 minutes to do it – it is OK to leave early if you are done.

• You should put your name on all test materials.

• It should be easy for me to find your answers. If you put them anywhere, but on the test itself, please include a note so that I can find it.

• The sample test is an approximation of the real test
  – There may be stuff on the real test that is not on the practice and vice versa
  – The real test will be no longer than the practice test – it may be shorter

• Strategy: Do all the fast questions first
  – Initially, do not spend more than 7 minutes on a question.
  – Then go back and complete what you didn't on the first round
Regexp = formula specifying set of strings

- Regexp = ∅
  - The empty set (base case 1, doesn't recognize any strings)

- Regexp = ε
  - The empty string (base case 2, recognizes the empty string)

- Regexp = a sequence of one or more characters from the set of characters
  - X
  - Y
  - This sentence contains characters like &T^**%P

- Disjunctions, concatenation, and repetition of regexps yield new regexps
Concatenation, Disjunction, Repetition

• Concatenation
  – If X is a regexp and Y is a regexp, then XY is a regexp
  – Examples
    • If $ABC$ and $DEF$ are regexps, then $ABCDEF$ is a regexp
    • If $AB^*$ and $BC^*$ are regexps, then $AB^*BC^*$ is a regexp
      – Note: Kleene * is explained below

• Disjunction
  – If X is a regexp and Y is a regexp, then X | Y is a regexp
  – Example: $ABC|DEF$ will match either $ABC$ or $DEF$

• Repetition
  – If X is a regexp than a repetition of X will also be a regexp
    • The Kleene Star: $A^*$ means 0 or more instances of $A$
    • Regexp{number}: $A\{2\}$ means exactly 2 instances of $A$
Regexp Notation Continued

- **Disjunction of characters**
  - \([ABC]\) – means the same thing as \(A\mid B\mid C\)
  - \([a-zA-Z0-9]\) – ranges of characters equivalent to listing characters, e.g., \(a\mid b\mid c\mid \ldots \mid A\mid B\mid \ldots \mid 0\mid 1\mid \ldots \mid 9\)
  - ^ inside of bracket means complement of disjunction, e.g., \([^a-z]\) means a character that is neither a nor b nor c … nor z

- **Parentheses**
  - Disambiguate scope of operators
    - \(A(BC)\mid (DEF)\) means ABC or ADEF
    - Otherwise defaults apply, e.g., \(ABC\mid D\) means \(ABC\) or \(ABD\)

- **? signifies optionality**
  - \(ABC?\) is equivalent to \((ABC)\mid (AB)\)

- **+ indicates 1 or more**
  - \(A(BC)\ast\) is equivalent to \(A\mid (A(BC)\ast)\)
Regexp Notation Continued

• Special Symbols:
  – $A.\ast B$ – matches A and B and any characters between (period = any character)
  – $^ABC$ – matches ABC at beginning of line (^ represents beginning of line)
  – $/[.?!]$/ – matches sentence final punctuation ($) represents end of line)

• What is a “good” regexp to solve some task
  – Not overly specific
    • Capture generalizations
    • Covers some unseen examples
  – Not overly general
    • Should not match obviously wrong cases
Sample Regular Expression

A regular expression for a date that will include the following expressions:

- January 3, 2012
- January, 2012
- January 3
- Jan. 3, 2012

Specific Constraints:

- Allow all the months of the year, as well as abbreviations that consist of the first 3 letters of the month and a period
- The day should be a one or two digit number
- The year should be a four digit number
- The day and year are optional
- A comma and a space precedes the year.

Bad answers

- Too Specific: \(January(3)\,(2012)|(January\,3)|(Jan\.\,3,\,2012)\)
- Too General: \([A-Za-z\.]+(\,[0-9])?,(\,[0-9]+)\)*

Good Answer: \(((Jan(uary)?)|(Feb(ruary)?)|(Mar(ch)?)|(Apr(il)?)|(May)|(Jun(e)?)|(Jul(y)?)|(Aug(ust)?)|(Sep(t?)|(ember)?)|(Oct(ober)?)|(Nov(ember)?)|(Dec(ember)?)\).?\([1-3]?[0-9]?\)?(, [0-9]{4})?
Phrase Structure Rules and Tree

• Draw a Phrase Structure Tree, including Penn POS tags, and List the Rules for the following sentence from Wikipedia (shortened slightly):
  – *Parodia tenuicylindrica is a small species of cactus native to Brazil*
  – We will assume that:
    • species names consist of proper nouns
    • all the words in proper noun phrases are proper nouns
  – This phrase *native to Brazil* is the most difficult part of this sentence to analyze and I may ask a specific question about this.
Parodia tenuicylindrica is a small species of cactus native to Brazil
Phrase Structure Rules used

• \( S \rightarrow \text{NP} \ \text{VP} \)
• \( \text{NP} \rightarrow \text{NNP} \ \text{NNP} \)
• \( \text{NP} \rightarrow \text{NNP} \)
• \( \text{NP} \rightarrow \text{NN} \)
• \( \text{NP} \rightarrow \text{DT ADJ NN} \)
• \( \text{NP} \rightarrow \text{NP PP ADJP} \)
• \( \text{PP} \rightarrow \text{IN NP} \)
• \( \text{VP} \rightarrow \text{VBZ NP} \)
• \( \text{ADJP} \rightarrow \text{JJ PP} \)
Things to Remember about Phrase Structure

• PP → P NP
  – *in the room, at the table, by John, with gusto*

• Sbar → special_word S
  – special_word → that, for, subord_conj, wh_word
  – *that she would leave soon*
  – *for her to leave*
  – *if she leaves*

• CC combines 2 or more Xs to produce a new X
  – *[NP [NP John] [CC and] [NP [DT the] [NN blender]]]*
  – *[S [S The ball went up] [CC or] [S Maybe it didn't]]*

• Punctuation matters – it should be included, typically with itself or PU as its POS
Chomsky Normal Form (required by CKY)

- Context Free Grammars can be converted to CNN
  - 3 types of rules:
    - XP → YP ZP  Binary Branching
    - XP → x  NonTerminal → terminal
    - XP → ε  NonTerminal → empty string

- Conversion
  - Replace \( VP \rightarrow VG \) & \( NP \rightarrow NG \) with nonbranching rules expanding VP and NP to whatever VG and NG matched to:
    - Not good examples: \( VP \rightarrow ate, VP \rightarrow had, NP \rightarrow food, \ldots \)
    - Usually V and N are assumed to be nonterminals, and the POS of words are looked up
  - Replace \( VP \rightarrow V NP PP \) with 2 rules:
    - \( VP \rightarrow VG PP \)
    - \( VG \rightarrow V NP \)
  - Replace: \( NP \rightarrow POSSP N PP \) with 2 rules:
    - \( NP \rightarrow NG PP \)
    - \( NG \rightarrow POSSP N \)
Sample Grammar for CKY

- \( S \rightarrow NP \ VP \)
- \( NP \rightarrow D \ N \)
- \( NP \rightarrow PossP \ N \)
- \( NP \rightarrow N \)
- \( PossP \rightarrow NP \ Poss \)
- \( VP \rightarrow V \ NP \)
- \( VP \rightarrow V \)
- \( N \rightarrow clam \)
- \( N \rightarrow edges \)
- \( N \rightarrow shell \)
- \( D \rightarrow the \)
- \( Poss \rightarrow 's \)
- \( V \rightarrow has \)
- \( V \rightarrow edges \)
- \( V \rightarrow shell \)
### CKY Parse Chart

<table>
<thead>
<tr>
<th></th>
<th>The</th>
<th>clam</th>
<th>'s</th>
<th>shell</th>
<th>had</th>
<th>edges</th>
</tr>
</thead>
<tbody>
<tr>
<td></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</td>
<td>NP</td>
<td>POSSP</td>
<td>NP</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>[0,1]</td>
<td>[0,2]</td>
<td>[0,3]</td>
<td>[0,4]</td>
<td>[0,5]</td>
<td>[0,6]</td>
</tr>
<tr>
<td>1</td>
<td>N, NP</td>
<td>POSSP</td>
<td>NP</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>[1,2]</td>
<td>[1,3]</td>
<td>[1,4]</td>
<td>[1,5]</td>
<td>[1,6]</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>POSS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[2,3]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>N,NP,V,VP</td>
<td>S</td>
<td>[3,5]</td>
<td>[3,6]</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>V, VP</td>
<td>VP</td>
<td>[4,5]</td>
<td>[4,6]</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>N,NP,V,VP</td>
<td>S</td>
<td>[5,6]</td>
<td></td>
</tr>
</tbody>
</table>
Viterbi Decoding of HMM for *rose pickles*

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

- **Transition Probabilities:**

![Diagram of transition probabilities for Viterbi Decoding of HMM for rose pickles](image)
Rose Pickles

- **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>Start</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NNP</td>
<td>.42 *.01</td>
<td><em>0</em>.001</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>*.001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NNS</td>
<td></td>
<td>*.03</td>
<td><em>.3</em>.03</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>.3</em>.03</td>
<td><em>.24</em>.03</td>
<td></td>
</tr>
<tr>
<td>NN</td>
<td>.20 *.02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VBZ</td>
<td></td>
<td></td>
<td><em>.3</em>.05</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>.3</em>.05</td>
<td><em>.1</em>.05</td>
<td></td>
</tr>
<tr>
<td>VBD</td>
<td>.05 *.05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End</td>
<td></td>
<td></td>
<td></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)
## Rose Pickles

- **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>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>
</tr>
<tr>
<td>NNP</td>
<td></td>
<td><strong>.42 * .01</strong></td>
<td><em><em>0</em>.001=0</em>*&lt;br&gt;<em><em>0</em>.001=0</em>*&lt;br&gt;<strong>.23*.001 = 9.67 * 10^-7</strong></td>
</tr>
<tr>
<td>NNS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN</td>
<td></td>
<td><strong>.20 * .02</strong></td>
<td></td>
</tr>
<tr>
<td>VBZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VBD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>End</strong></td>
<td>0</td>
<td><strong>.05 * .05</strong></td>
<td><strong>.2 = 1.93 * 10^-7</strong>&lt;br&gt;<strong>.15 = 5.4*10^-6</strong>&lt;br&gt;<strong>.45 = 2.84*10^-5</strong></td>
</tr>
</tbody>
</table>
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
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?
    \[
    \frac{0+0+0+(34\times4)+(0.8\times5)}{\sqrt{(0+0+0+34^2+8^2) \times (1^2+2^2+3^2+4^2+5^2)}} = \frac{140}{\sqrt{640 \times 55}} \approx 0.2014
    \]
Sequence Labeling with BIO tags

• Noun group BIO tags
  – The B
  – big I
  – bad I
  – wolf I
  – approached 0
  – the B
  – house I

• NE BIO tags
  – However 0
  – , O
  – International B-ORG
  – Business I-ORG
  – Machines I-ORG
  – and O
  – Google B-ORG
  – rose 0
  – in 0
  – active 0
Basic NE types

• Person – a person name (*Mary Smith*) or a set of people (*the Smith family*)
• GPE – Name associated with land mass, a government and the people who live there (*the United States, New Jersey, …*)
• ORG – Name associated with a company, club, or other type of structured unit with members, employees and/or other types of participants (*IBM, the Catholic Church, the NY Police Department, …*)
Binding Theory for English 3rd Pers Prons

• Case 1: If the pronoun $p$ is inside an NP premodified by a possessive, the antecedent needs to be outside of this NP
  
  – *John* likes *Mary's drawing of him*
  
  – *John* likes *his drawing of Mary*

• Case 2: Otherwise, the antecedent must be outside the immediate tensed clause containing the personal pronoun.
  
  – *John* said that *he* liked pizza.
  
  – *John* wanted for *him* to like pizza.
  
  – *John* liked *him*.

• Theories of binding vary about how these (and similar) constraints are encoded, but the differences in the final result (quality of system output) is minimal. While these 2 rules cover most cases, there are also some exceptions:
  
  – *John* always carries a slice of pizza with *him*. 
Binding Theory for English Reflexives/Reciprocals

• The antecedent of a reflexive/reciprocal must be the closest subject or possessive such that:
  – The antecedent precedes and “commands” the pronoun
    • A commands B if A is the sibling of a phrase that dominates B.
  – There is no possessive or subject for phrases in the path in the phrase structure tree between antecedent and pronoun

• Examples:
  – Mary saw herself vs. *Mary said that John would meet herself soon
  – Mary's picture of herself vs. *Mary saw John's picture of herself

• These rules covers most cases.
  – Exception: Pictures of themselves made the actors nervous.
Hobbs Search Algorithm to Find Antecedent of Anaphors

1. Go to NP immediately dominating pronoun
2. Go up to 1st dominating NP or S node. This node = X path to X = p.
3. Traverse branches below X to the left of p, left-to-right and and breadth first. Propose each NP n as an antecedent if there is an NP or S between n and X.
4a. Is X the highest S in the sentence? (Recursive)
   - Yes
   - No
5. From Node X, go up to the 1st NP or S. Call this node X and the path to X p.
6. If X = NP and there is no N’ in p, propose X as antecedent
7. Search for antecedent in branches below X, left-to-right, breadth first. Propose each NP.
8. If X is an S node, traverse all branches of X preceding p, left-to-right, breadth first, but not going below any S or NP node found. Propose each NP.
End

End
Hobbs Search Example

1. Mary saw the chicken.

2. Jim said that she laughed.
Evaluation

• Accuracy: If the answer key and system output are guaranteed to be the same length
  \[ \text{Accuracy} = \frac{\text{Correct}}{\text{Total Items}} \]

• Recall/Precision/F-Score given answer key for a task
  \[ \text{Recall} = \frac{|\text{Correct}|}{|\text{Answer Key}|} \quad \text{Precision} = \frac{|\text{Correct}|}{|\text{System Output}|} \quad \text{F-Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} \]

• Kappa – How much better is interannotator agreement than chance?
  \[ \text{Kappa} = \frac{\text{Percent (Actual Agreement)} - \text{Prob (Chance Agreement)}}{1 - \text{Prob (Chance Agreement)}} \]

• Coreference with the B-cubed algorithm
  – Precision: For each partition of Coreferential NPs in system output: average the percentage of links in the answer
  – Recall: For each partition of Coreferential NPs in the answer, average the percentage of links in the system output
  – Example:
    • 2 system entities: \{A_1, A_2, A_3, A_4, C_1\} and \{B_1, B_2\}
    • 2 answer key entity: \{A_1, A_2, A_3, A_4, B_1, B_2\} and \{C_1\}
    • Precision = (4 X 4/5 + 1/5 + 2) X 1/7 = .77
    • Recall = ((4 X 2/3) + (2 X 1/3) + 1) X 1/7 = .62
Example Feature Structure Question

- What (Typed) Feature Structure would result if FS1 was unified with the value of the path *quant* in FS2?

  - Note: *orth* is short for *orthography* and *cat* is short for *category*
Answer to Sample FS Question

[Diagram showing a linguistic analysis structure with nodes labeled as follows:

- Quant
- Determiner
- These
- Head
- Agreement (repeated multiple times)
- Number
- Person
- Plural
- 3rd
- Orth
- N
- Fish
- DT
- Cat
- Orth]

Computational Linguistics
Review Lecture
2016
Machine Translation Questions

• Decoding Question (like the sample test)
• Walk through part of Maximization/Estimatation
  – Given a set of translation probabilities, calculate the probability of a particular alignment
  – Given a set of alignments and their probability scores, calculate the new translation probabilities
Simplified Example of EM model

• Given
  – 4 French words: *la*, *maison*, *bleu*, and *fleur*
  – 4 English words: *the*, *house*, *blue* and *flower*
  – We only allow 1 to 1 alignments

• Starting assumption
  – Each French word has a .25 chance of being translated as a given English word
Initial Alignment Probs for 3 E/F pairs

- Initial: la/the (.25), maison/the (.25), la/house (.25), maison/house (.25)

- la maison $\rightarrow$ the house
  - la/the $\times$ maison/house = .0625
  - maison/the $\times$ la/house = .0625

- la maison bleu $\rightarrow$ the blue house (all possible alignments)
  - la/the $\times$ maison/house $\times$ bleu/blue = $0.25^3 = 0.015625$
  - la/the $\times$ maison/blue $\times$ bleu/house = .015625
  - la/house $\times$ maison/the $\times$ bleu/blue = .015625
  - la/house $\times$ maison/blue $\times$ bleu/house = .015625
  - la/blue $\times$ maison/house $\times$ bleu/the = .015625
  - la/blue $\times$ maison/the $\times$ bleu/house = .015625

- La fleur $\rightarrow$ the flower
  - la/the $\times$ fleur/flower = .0625
  - fleur/the $\times$ la/flower = .0625
Maximum Liklihood Estimates (MLE)

- For each e/f pair and for each sentence, add up the probabilities of alignments that contain that pair and regularize to 1 (initially: all prob=.25)
- Sum these scores and divide by the number of instances of f.
- Translations from X to **the**
  - *la/the*: .5 of the first set of alignments, .33 of the second set and .5 of the 3\(^{rd}\)
    - \(\frac{(.5 + .33 + .5)}{3} = .44\)
  - *maisson/the*: .5 of the 1\(^{st}\) + .33 of the 2\(^{nd}\), 0 in the 3\(^{rd}\)
    - \(\frac{(.5 + .33)}{3} = .42 = .29\)
  - *bleu/the*: 0 in the 1\(^{st}\) + .33 of the 2\(^{nd}\) + 0 in the 3\(^{rd}\)
    - \(\frac{.33}{3} = .11\)
  - *fleur/the*: 0 in the 1\(^{st}\) and 2\(^{nd}\), .5 in the 3\(^{rd}\)
    - \(\frac{.5}{3} = .17\)
- *house*: *la/house*=.42, *maisson/house*=.42, *bleu/house*=.17, *fleur/house*=0
- *blue*: *la/blue*=.33, *maisson/blue*=.33, *bleu/blue*= .33, *fleur/blue*=0
- *flower*: *la/flower*=.5 *maisson/flower*=0, *blue/flower*=0, *fleur/flower*= .5
Expectation: Rescore Alignments

• \textit{la maison} \rightarrow \textit{the house}
  – \textit{la/the} (.44), \textit{maisson/the} (.29), \textit{la/house} (.42), \textit{maisson/house} (.42)
  – \textit{la/the} X \textit{maisson/house} = .1848
  – \textit{maisson/the} X \textit{la/house} = .1276

• \textit{la maison bleu} \rightarrow \textit{the blue house} (all possible alignments)
  – \textit{la/the} X \textit{maisson/house} X \textit{bleu/blue} = .06098
  – \textit{la/the} X \textit{maisson/blue} X \textit{bleu/house} = .02468
  – \textit{la/house} X \textit{maisson/the} X \textit{bleu/blue} = .04019
  – \textit{la/house} X \textit{maisson/blue} X \textit{bleu/house} = .02356
  – \textit{la/blue} X \textit{maisson/house} X \textit{bleu/the} = .045274
  – \textit{la/blue} X \textit{maisson/the} X \textit{bleu/house} = .016269

• \textit{La fleur} \rightarrow \textit{the flower}
  – \textit{la/the} X \textit{fleur/flower} = .22000
  – \textit{fleur/the} X \textit{la/flower} = .08500
Translating sample sentence

• Input: *La maissan bleu*
• Translation probabilities (hypothetical):

<table>
<thead>
<tr>
<th>French</th>
<th>English</th>
<th>Unigram probabilities (count in WSJ ÷ 1 million)</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>la</em></td>
<td><em>the</em></td>
<td><em>the</em> = .035, <em>blue</em> = 1.3 X 10^-4, <em>house</em> = 6.7 X 10^-4, <em>flower</em> = 6 X 10^-6</td>
</tr>
<tr>
<td><em>maisson</em></td>
<td><em>the</em></td>
<td></td>
</tr>
<tr>
<td><em>bleu</em></td>
<td><em>blue</em></td>
<td></td>
</tr>
<tr>
<td><em>fleur</em></td>
<td><em>flower</em></td>
<td></td>
</tr>
</tbody>
</table>

• The most probable translation would be:

  – *the house blue* = translation-prob X language prob = 4.37 X 10^-10
  • translation-prob = .7 X .5 X .41 = .1435
  • Lang-prob = .035 X 6.7 X 10^-4 X 1.3 X 10^-4 = 3.05 X 10^-9
Sample Annotation Task

• Semantic Role Labeling: Find each noun like ABILITY (listed in allcaps) and mark 2 arguments: a sentient NP that can do an action (ARG0) and a description of the action or type of action (ARG1). ARG0 and ARG1 are both optional. Arguments can occur anywhere in the sentence, but prefer close arguments.

• Examples:
  – [the government]'s ABILITY [to pay its bills]
    ARG0                                       ARG1
  – [the government]'s legal CAPACITY
    ARG0                                       ARG1
Data to Mark Up

• The government's borrowing AUTHORITY declined at midnight on Tuesday
• Mr. Honecker headed the Poliburo's security APPARATUS
• The complicated new funding DEVICE will cause more problems than it will solve.
• The First World has for some time had the bad HABIT of smothering other people's economies with this kind of unfocused kindness.
• So do just about all the losses that could be attributed to the sheer INCOMPETENCE of unqualified planners.
Practice Final and Answers