Announcement

- Midterm Thursday or next week on Tuesday?

Question 4: Linguistics

We have discussed various levels of linguistic structure, including orthography, phonetics, morphology, syntax, and pragmatics. Each sentence below violates the standard rules of English on one of these levels. Label each sentence accordingly.

- I like marvel cat.
- I like cars many.
- Q: Would you mind passing the cat? A: No, I wouldn't. (nothing happens)
Question 1: PCFG's

Nathan L. Pedant decides to build a treebank. He finally produces a corpus which contains the following three parse trees:

Clarissa Leicina then produces the treebank, and decides to build a PCFG, and a parser, using Nathan's data.

Question (a): Show the PCFG that Clarissa would derive from this treebank, using a maximum likelihood estimator.

Question (b): Show two parse trees for the string "Jeff pronounced that Fred scored loudly", and calculate their probabilities under the PCFG.

Question (c): Clarissa is shocked and dismayed, (see (b)), that "Jeff pronounced that Fred scored loudly" has two possible parses, and that one of them—that Jeff is doing the pronouncing loudly—has relatively high probability, in spite of it having the ADVP clausally modifying the "higher" verb, pronounced. This type of high attachment is never seen in the corpus, so the PCFG is clearly missing something. Clarissa decides to fix the treebank, by altering some non-terminal labels in the corpus. Show one such transformation that results in a PCFG that gives zero probability to parse trees with "high" attachments. (Your solution should systematically re-label some non-terminals in the treebank, in a way that slightly increases the number of non-terminals in the grammar, but allows the grammar to capture the distinction between high and low attachment to VPs.)

Question 2: Trigram HMM's

In class we discussed the trigram HMM model for part-of-speech tagging. Assume that we have a vocabulary of $V$ words and a set of $K$ part-of-speech tags. We define the joint probability of a sentence and its tags as follows:

$$P(w_t, t_t) = P(\text{STOP})_t \cdot \prod_{i=1}^{L-1} P(t_i|t_{i-1})P(w_i|t_i)$$

where $w$ and $t$ are word and tag sequences, respectively, both of length $L$. Given an HMM and a sentence $w$, we would like to efficiently compute $\arg\max_t P(w|t)$.

Question (a): How many possible tag sequences exist for this sentence? Is it feasible to separately compute the probability of each?

Question (b): Fortunately, the Viterbi algorithm allows us to efficiently find $\arg\max_t P(w|t)$. How does this tag sequence relate to the desired tag sequence $\arg\max_t P(w|t)$? Justify your answer.

Question (c): The Viterbi algorithm is an example of a dynamic program. In one short sentence, describe the intuition behind Viterbi: how it allows us to find the highest probability sequence from among an exponential number of possibilities in polynomial time.
Question (6d): Now let’s consider a tagger HMM:

\[ P(x_t) = P(\text{STOP}) q_0 \cdot \prod_{t=1}^{T} P(x_{t-1}, x_t) P(w_t | x_t) \]

There are two ways to modify our set-up account for this. One way involves artificially increasing the tag-set, leaving the Viterbi algorithm unchanged. The other way involves keeping the tag-set unchanged, but altering Viterbi. Describe these two methods, using pseudo-code or recursive equations where appropriate. What are the new time and space complexities of these two methods?

---

**Viterbi Equations**

- **max prob of**
- **being in state \( q_0 \)**
- **at time \( t \)**

\[
V[0, q_0] = 1 \\
V[t, q_j] = \max_{q_i} V[t-1, q_i] \times a_{i,j} \times b_{j, w_t} \\
goal = \max_{q_i} V[n, q_i] \times a_{i, f}
\]

**Viterbi Pseudocode**

- \( V[*, \cdot] \leftarrow 0 \)
- \( V[0, q_0] \leftarrow 1 \)
- for \( t = 1 \ldots n \)
  - for \( j = 1 \ldots |Q| \)
    - for \( i = 1 \ldots |Q| \)
      - \( V[t, q] \leftarrow \max\{ V[t, q], V[t-1, q_i] \times a_{i,j} \times b_{j, w_t} \} \)
- for \( i = 1 \ldots |Q| \)
  - \( V[n+1, q] \leftarrow \max\{ V[n+1, q], V[n, q] \times a_{i} \} \)
- return \( V[n+1, q_0] \)
Bill directed plays about English kings.

\[ V[t,q_t] = \max_{q_{t-1}} V[t-1,q_{t-1}] \times a_{i,j} \times b_{j,u_i} \]

**Probabilistic CKY**

\[
\begin{align*}
C[i\ldots i, x_i] &= 1 \\
C[i\ldots i, V] &= p(V \rightarrow x_i) \\
C[i\ldots k, V] &= p(V \rightarrow YZ) \times C[i\ldots j, Y] \times C[j\ldots k, Z] \\
p(c^* | \{x_1\ldots x_n\}) &= C[0, n, \mathcal{S}] 
\end{align*}
\]
Probabilistic CKY Pseudocode

• Input: x, G
• Output: C
• for i = 1 ... n
  • C[A, i-1, i] = p(A → x)
• for ℓ = 2 ... n
  • for i = 0 ... n - ℓ
    • k = i + ℓ
    • for j = i + 1 ... k - 1
      • C[X, i, k] = max(C[X, i, k], p(X | X → YZ) × C[Y, i, j] × C[Z, j, k])
  • return C[S, 0, n]

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]
Question 3: Interpolated Language Models

In class we discussed a language model which interpolates between unigrams, bigrams, and trigrams:

\[ P_{\text{merge}}(w) = \prod_i \lambda_i P_{\text{unigram}}(w_i) + \lambda_2 P_{\text{bigram}}(w_i|w_{i-1}) + \lambda_3 P_{\text{trigram}}(w_i|w_{i-1}, w_{i-2}) \]

Where \( \lambda_1, \lambda_2, \lambda_3 \) are positive values that sum to 1, and are either fixed by hand or learned via EM.

Question (a): For which kind of training corpus would you want to place higher relative weight on the unigram model? Briefly justify your answer.

Question (b): For which kind of training corpus would you want to place higher relative weight on the unigram model? Briefly justify your answer.

Question (c): Now consider the possibility that we allow \( \lambda_1, \lambda_2, \lambda_3 \) to vary according to the particular trigram that is being computed. Write down the functional form for this new model \( P_{\text{merge}}(w) \). Hint: \( \lambda_1, \lambda_2, \lambda_3 \) must now be considered functions rather than fixed values.

Question (d): For which trigrams would we want to place higher weight on the trigrams vs. unigrams submodels? Briefly justify your answer.