

More on NLP

- An excerpt from Games magazine, November 2005

"The only smurf we have to smurf is smurf itself."

"Yea, though I smurf through the smurf of the smurf of smurf, I will smurf no smurf."

"The first smurf about Smurf Smurf is: you do not smurf about Smurf Smurf. The second smurf about Smurf Smurf is: you do NOT smurf about Smurf Smurf."



- What quotes do these lines correspond to?
- How were you able to figure out the meanings of these words, when half of the words have been replaced?

CSC384 Lecture Slides © Steve Engels, 2005

Slide 1 of 30

Another Language Example

- From *Jabberwocky* (a poem in *Through the Looking-Glass and What Alice Found There*), by Lewis Carroll, 1872:

"Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogroves
And the mome raths outgrabe.



- What is the part-of-speech tag for "brillig", "gimble" and "borogroves"?
- What's the present tense for "outgrabe"?
- How do we know what POS tags to use?

CSC384 Lecture Slides © Steve Engels, 2005

Slide 2 of 30

Statistical NLP

- Using CFGs and parsing is a result of Noam Chomsky's linguistic influence, which assumes that humans speech is based on these rule-based systems
- Statistical NLP is partly concerned with supplementing CFGs with probabilistic weights, but also with **stochastic models** of language
 - rationalist vs. empiricist views of language
- Stochastic models are based on data
 - word frequency (lexicon)
 - automatic rule-forming (grammar)

CSC384 Lecture Slides © Steve Engels, 2005

Slide 3 of 30

Statistical NLP at Work

- The **attachment ambiguity** problem:

"The child ate the ice cream with a spoon."

- Does the prepositional phrase (PP) attach itself to the verb, or to the object of the sentence?
- Problem can be solved by examining other instances of that prepositional phrase, given the previous cues:

| word | C(word) | C(word, with) | C(with word) |
|-----------|---------|---------------|--------------|
| ate | 5156 | 607 | 0.118 |
| ice cream | 1442 | 155 | 0.107 |

- Gives even more certainty if all the preposition phrase words and/or the total sentence structure are included in the calculation

CSC384 Lecture Slides © Steve Engels, 2005

Slide 4 of 30

Lexical Resources

- In order to build a model, one needs machine-readable text, dictionaries, thesauri and tools for processing them
- Hand-tagged corpora (plural of corpus, Latin for "body") are essential
 - Brown corpus**: balanced corpus, ~1 million words
 - Penn Treebank**: over 1 million words, with syntactic structure
 - Canadian Hansards**: bilingual proceedings of Parliament
 - WordNet**: dictionary with words organized into **synsets**
- Zipf's Law**: if the frequency and rank of words in a corpus are measured, then:

$$f \propto 1/r$$

CSC384 Lecture Slides © Steve Engels, 2005

Slide 5 of 30

Lexical Terms

- Collocations** = a turn of phrase or grouping of words whose whole has a significance beyond the sum of its parts (e.g. disk drive, make up, to and fro, New York)
- Concordances** = the connection between a single word and the other words in its surrounding context
 - KWIC** (Key Word In Context) program: special software that displays the concordances of a word in text
- Morphology** = the study of modifying words during the lexical processing stage
 - grouping similar words together (organize, organizes) through **stemming/lemmatization**
 - separation of differing concepts (organize, organization)

CSC384 Lecture Slides © Steve Engels, 2005

Slide 6 of 30

n-gram Models

- Statistical NLP concerns itself with modeling sequences of words, instead of decompositions of sentences
- The **n-gram** model tries to predict the likelihood of a word, given the past observations in the sentence:

$$P(w_n | w_1, \dots, w_{n-1})$$

- **Markov assumption** = only the last few words really affect what the next word will be
 - typically, people use **bigram** or **trigram** language models
 - helps deal with sparse data issues...sort of

Maximum Likelihood Estimation

- Maximum Likelihood Estimation (MLE) estimates from relative frequencies of words in the training text
- Given text of N words, by appending n-1 dummy start symbols to the text, we can say that the corpus contains N n-grams
- If $C(w_1, \dots, w_n)$ is the frequency of a certain n-gram,

$$P_{MLE}(w_1, \dots, w_n) = \frac{C(w_1, \dots, w_n)}{N}$$

$$P_{MLE}(w_n | w_1, \dots, w_{n-1}) = \frac{C(w_1, \dots, w_n)}{C(w_1, \dots, w_{n-1})}$$

n-gram Problems

- With higher values for n, one would be able to estimate the next word with greater reliability
- The problem as always, is sparse data.
 - higher values of n decrease the probability that a particular n-gram has been seen before
 - Example: trigram model in Jane Austen writings
 - Given the words "to both", what is the next word?
 - This trigram occurs 9 times total in the corpus, and only one word occurs more than once – "to"
 - four-gram (or tetragram) models are practically useless in these cases
- Common solutions:
 - use higher order n-gram models only in cases of plentiful data
 - treat all unseen n-grams as having a minimal likelihood, thus adding 1 to the numerator of the MLE calculation (Laplace's Law)

Word Sense Disambiguation

- Similar to the estimation of the next word is the understanding of the meaning of a current word
- **Bayes classifier**:
 - given word w_i with context c, how do we assign it a sense s' ?

$$P(s' | c) > p(s_k | c) \text{ for } s_k \neq s'$$

$$s' = \operatorname{argmax}_{s_k} \frac{P(c | s_k) P(s_k)}{P(c)}$$

- The Naive Bayes assumption declares that:

$$P(c | s_k) = P(\{v_j | v_j \text{ in } c\} | s_k) = \prod_{v_j \text{ in } c} P(v_j | s_k)$$

- **Translation**: determine the sense of most of the words that fit into that context

Alternate WSD Techniques

- Dictionary/Thesaurus disambiguation:
 - using a dictionary that outlines possible word senses, examine the context to see whether the surrounding words match one definition's domain or the other's
 - Example: "My *bank* branch is on the *bank* of the Thames"
- Translation-based disambiguation:
 - given a bilingual corpus (Hansards, e.g.), one can find an example of the given phrase in one, and determine the word sense more precisely in the other language
 - Example: "earn interest" versus "show interest"
 - translated into German, the first phrase becomes "Beteiligung erwerben", whereas the second phrase becomes "Interesse zeigen", both of which have clear word senses in German.

Selectional Preferences

- Given a word that we haven't seen before, how do we tell what semantic role it has?
 - Example: "Eugen had never eaten a fresh *durian* before."
- "Preference" is used here instead of "rule", since the object in question might not be the food item that we would assume (e.g. "eat one's words")
- Classification technique would be similar to the task of word sense disambiguation, adapted to the task of obtaining semantic information instead of just syntactic information.



Lexical Acquisition

- The idea of selectional preferences can be used to acquire lexical information
 - lexical information is often sparse or difficult to tag by hand
 - automated tagging can save time and manpower by reducing the tagging task to a proofreading one
- Acquiring these groups involves a **clustering** algorithm
- k-nearest neighbour** (kNN) algorithm:
 - assuming n classification categories, pick n random contexts for a word in a sentence
 - find the k words that fit the closest into that context
 - change the category to be the average context for those k words, and repeat until steady state is reached

CSC384 Lecture Slides © Steve Engels, 2005

Slide 13 of 30

Transformation-Based Tagging

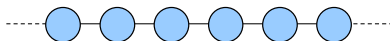
- Another variation on the use of context in POS tagging is Brill's **transformation-based** tagger:
 - made up of triggering environment and rewrite rules
- General algorithm:
 - All words are tagged with the most likely part-of-speech
 - Triggers are evaluated one at a time. If a triggering environment is satisfied, then the part-of-speech is rewritten for the word that activated the trigger
 - This continues until all the triggers have been evaluated. The part-of-speech for certain rules may be rewritten multiple times by successive rules
 - rules with highest precedence are left to the end of the list
- Learning algorithm for this tagger involved reordering the trigger/rewrite list until the transformations produce the most accurate tagging result

CSC384 Lecture Slides © Steve Engels, 2005

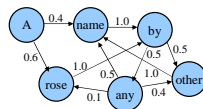
Slide 14 of 30

Markov Models

- Markov chains** represent a sequence of linked states as a single-chain graph:



- Markov models** are an adaptation of Markov chains, representing possible sequences of linked events, (also known as Visible Markov Models):

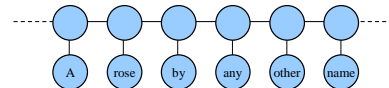


CSC384 Lecture Slides © Steve Engels, 2005

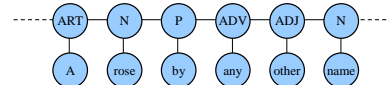
Slide 15 of 30

Hidden Markov Models

- Hidden Markov Models** (HMMs) represent a sequence of underlying states, each of which emits a token in the form of a word from the sentence.



- Often interested in most likely underlying POS values:



CSC384 Lecture Slides © Steve Engels, 2005

Slide 16 of 30

Hidden Markov Models (cont'd)

- Markov models have two basic characteristics
 - Limited Horizon:

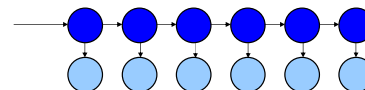
$$P(X_{t+1}=s_k | X_1, \dots, X_t) = P(X_{t+1}=s_k | X_t)$$
 - Time Invariance:

$$P(X_{t+1}=s_k | X_1, \dots, X_t) = P(X_2=s_k | X_1)$$
- Profound HMM thought:
 - "The past is independent of the future, given the present"
- Note: underlying states do not necessarily correspond to POS tags; POS tags are just a useful application

CSC384 Lecture Slides © Steve Engels, 2005

Slide 17 of 30

Defining HMM States

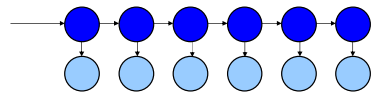


- The dark circles are the **hidden states** of the model
 - dependent only on the previous state
 - $S: \{s_1, \dots, s_N\}$
- The light circles are the **observed states**
 - depends on their corresponding hidden state
 - $K: \{k_1, \dots, k_M\}$

CSC384 Lecture Slides © Steve Engels, 2005

Slide 18 of 30

Defining HMM Probabilities



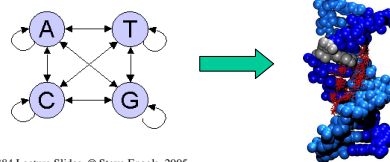
- The probabilities of each state occurring first are called the **initial state probabilities**
 - $\Pi: \{\pi_i\}$
- The probabilities between hidden states are called the **state transition probabilities**
 - $A = \{a_{ij}\}$
- The probabilities from hidden to observed states is called **observation state or emission probabilities**
 - $B: \{b_{jk}\}$
- HMM is defined as: $\{S, K, \Pi, A, B\}$

CSC384 Lecture Slides © Steve Engels, 2005

Slide 19 of 30

HMM Inferencing Tasks

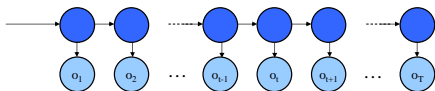
1. Compute the probability of a given observation sequence
2. Given an observation sequence, compute the most likely hidden state sequence
3. Given an observation sequence and set of possible models, which model most closely fits the data?



CSC384 Lecture Slides © Steve Engels, 2005

Slide 20 of 30

HMM Task #1: Decoding

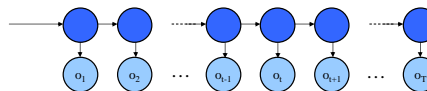


- Given an observation sequence:
 - $O: (o_1, \dots, o_T)$
- and a model:
 - $\mu: \{A, B, \Pi\}$
- Compute $P(O|\mu)$

CSC384 Lecture Slides © Steve Engels, 2005

Slide 21 of 30

Decoding (cont'd)



- Assuming state sequence X :

$$P(O | X, \mu) = b_{x_1 o_1} b_{x_2 o_2} \dots b_{x_T o_T}$$

$$P(X | \mu) = \pi_{x_1} a_{x_1 x_2} a_{x_2 x_3} \dots a_{x_{T-1} x_T}$$

$$P(O, X | \mu) = P(O | X, \mu) P(X | \mu)$$

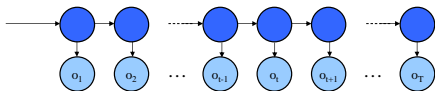
$$P(O | \mu) = \sum_X P(O | X, \mu) P(X | \mu)$$

$$P(O | \mu) = \sum_{\{x_1, \dots, x_T\}} \pi_{x_1} b_{x_1 o_1} \prod_{t=1}^{T-1} a_{x_t x_{t+1}} b_{x_{t+1} o_{t+1}}$$

CSC384 Lecture Slides © Steve Engels, 2005

Slide 22 of 30

Forward Procedure



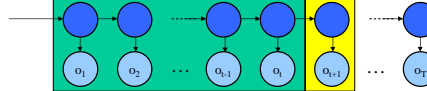
- Use dynamic programming to simplify HMM calculation.
- Intuition:
 - calculating the probability of sequences of $t+1$ observations involves the same calculation of sequences of t observations
 - Define:

$$\alpha_i(t) = P(o_1 \dots o_t, x_t = i | \mu)$$
 - $\alpha_i(t)$ is the probability of arriving at state i at time t

CSC384 Lecture Slides © Steve Engels, 2005

Slide 23 of 30

Forward Procedure (cont'd)



- Assuming we've calculated $\alpha_i(t)$, we can calculate $\alpha_j(t+1)$ without reproducing the $\alpha_i(t)$ calculation
- The probability of reaching state j at time $t+1$ depends on:
 1. the probability of reaching all possible states i at time t ,
 2. the probability of making the transition from i to j , and
 3. the probability of emitting observation o_{t+1}

$$\alpha_j(t+1) = \sum_{i=1, \dots, N} \alpha_i(t) a_{ij} b_{j o_{t+1}}$$

CSC384 Lecture Slides © Steve Engels, 2005

Slide 24 of 30

Backward Procedure

- Similar to forward procedure, except calculating probability of future state sequence, from state i at time t

$$\beta_i(t) = \sum_{j=1 \dots N} a_{ij} b_{io} \beta_j(t+1)$$

CSC384 Lecture Slides © Steve Engels, 2005 Slide 25 of 30

Combination Approach

$$P(O | \mu) = \sum_{i=1}^N \alpha_i(T)$$

$$P(O | \mu) = \sum_{i=1}^N \pi_i \beta_i(1)$$

↓

$$P(O | \mu) = \sum_{i=1}^N \alpha_i(t) \beta_i(t)$$

- When calculating the probability of a sequence, the combined approach allows one to perform the calculation from both directions
- The previous equations are special cases of this general one

CSC384 Lecture Slides © Steve Engels, 2005 Slide 26 of 30

Best State Sequence

- Determining the overall probability of a sequence is good, but finding the most likely underlying state sequence is better.
- Need to determine:

$$\arg \max_X P(X | O)$$

- Given M possible states and a sequence of N words, computing the most likely sequence could be $O(M^N)$ in theory
 - more efficient calculation: the **Viterbi** algorithm

CSC384 Lecture Slides © Steve Engels, 2005 Slide 27 of 30

The Viterbi Algorithm

- Basic algorithm:
 - advance through the state sequence, one stage at a time
 - at each stage, record the probability of the most likely path to reach each state
 - most likely path to any successive state X_i is determined by calculating the highest combined probability of a current state and its state transition to X_i
- Main principle:
 - uses dynamic programming to calculate the most probable path through the entire trellis
 - when recording the probability of the most likely path to a given state, also record the path itself for postprocessing
 - Viterbi algorithm finds best state sequence in $O(MN)$ time

CSC384 Lecture Slides © Steve Engels, 2005 Slide 28 of 30

The Viterbi Algorithm

- Calculating the most likely path to any of the second ("A") states of the sequence involves the most likely path to the preceding ("C") states.

CSC384 Lecture Slides © Steve Engels, 2005 Slide 29 of 30

Parameter Estimation

- The third and most complicated task
- Already showed how to calculate the overall probability of an emission sequence given the model, and how to determine the most likely state sequence, given an emission sequence and a model
 - the emission sequence is the set of words we observe
 - where did we get the model's transition and emission probabilities?
- Problem: how to learn the model parameters, given training data

CSC384 Lecture Slides © Steve Engels, 2005 Slide 30 of 30