Language and Computation

week 5, Tuesday, February 11, 2014

Tamás Biró
Yale University
tamas.biro@yale.edu
http://www.birot.hu/courses/2014-LC/

Practical matters

- **Background reading:** JM 7 and 8 (know what is in there).
- **Python:** this week H 5.
- **Sections:** probability theory
- Problem set 2 posted

Today

- Probability and smoothing

- From n-grams to Markov models

Next time:

Remarks on assignment 1

- Months of different length:
  `/(0[1-9]|1[0-2])\/(0[1-9]|[12][0-9]|3[01])/`

  `/((0[13578]|1[02])\/(0[1-9]|[12][0-9]|3[01])|`
  `(0[469]|11)\/(0[1-9]|[12][0-9]|3[0]))|`
  `(02)\/(0[1-9]|[12][0-9]))/`

- Leap years in Julian calendar: `\d\d([02468][048]|[1359][26])`

- Leap years in Gregorian calendar:
  `\d\d(0[48]|[2468][048]|[1359][26])|`
  ([02468][048]|[1359][26])00

Remarks on assignment 1

Finite-state technology:

- Finite number of states
- No "memory": no infinite memory
- Technically speaking: any large finite memory
- Is the phenomenon "inherently" finite-state?
  E.g., `01:January`. Reduplicative morphology.
  M. translation (Eng to Fr, Sp, Heb, Arb etc.): `Adj N` $\rightarrow$ `N Adj`

The “probability” of a document in language L?

Basics of probability

- **Sample space**: possible outcomes of an experiment.
- **Event**: a subset of the sample space.
- Given set X of *events* (a *random variable*)),
- **probability** $P : X \to [0, 1]$.
- $P(X) = 1$, $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$
- Conditional probability: $P(A|B) = P(A \wedge B)/P(B)$.

From frequency to probability to scores

What is "probability" P?

- Observed frequency in the training set/corpus?
- Expected frequency in the test set/corpus?
- Expected frequency in the "entire" set/corpus X?
- A technicality that sums up to 1?

Smoothing (overview only)

- N: corpus size (# of tokens)
  V: vocabulary size (# of types)
  c_i: count of word (type) w_i.

- Unsmoothed **Maximum Likelihood Estimate:**

$$P(w_i) = \frac{c_i}{N}$$

Smoothing (overview only)

- N: corpus size (# of tokens)
  V: vocabulary size (# of types, including those with 0 frequency!)
  c_i: count of word (type) w_i.

- **Laplace Smoothing** or **add-one smoothing:**

$$P_L(w_i) = \frac{c_i + 1}{N + V}$$

- as if we used $c_i^* = (c_i + 1)\frac{N}{N+V}$ in MLE,
  discounting c_i and reallocating probability mass to unseen words.

Smoothing (overview only)

- N: corpus size (# of tokens), c_i: count of word (type) w_i
  N_c: # of types that occur c times (frequency of frequency)
- **Good-Turing Smoothing/discounting**:

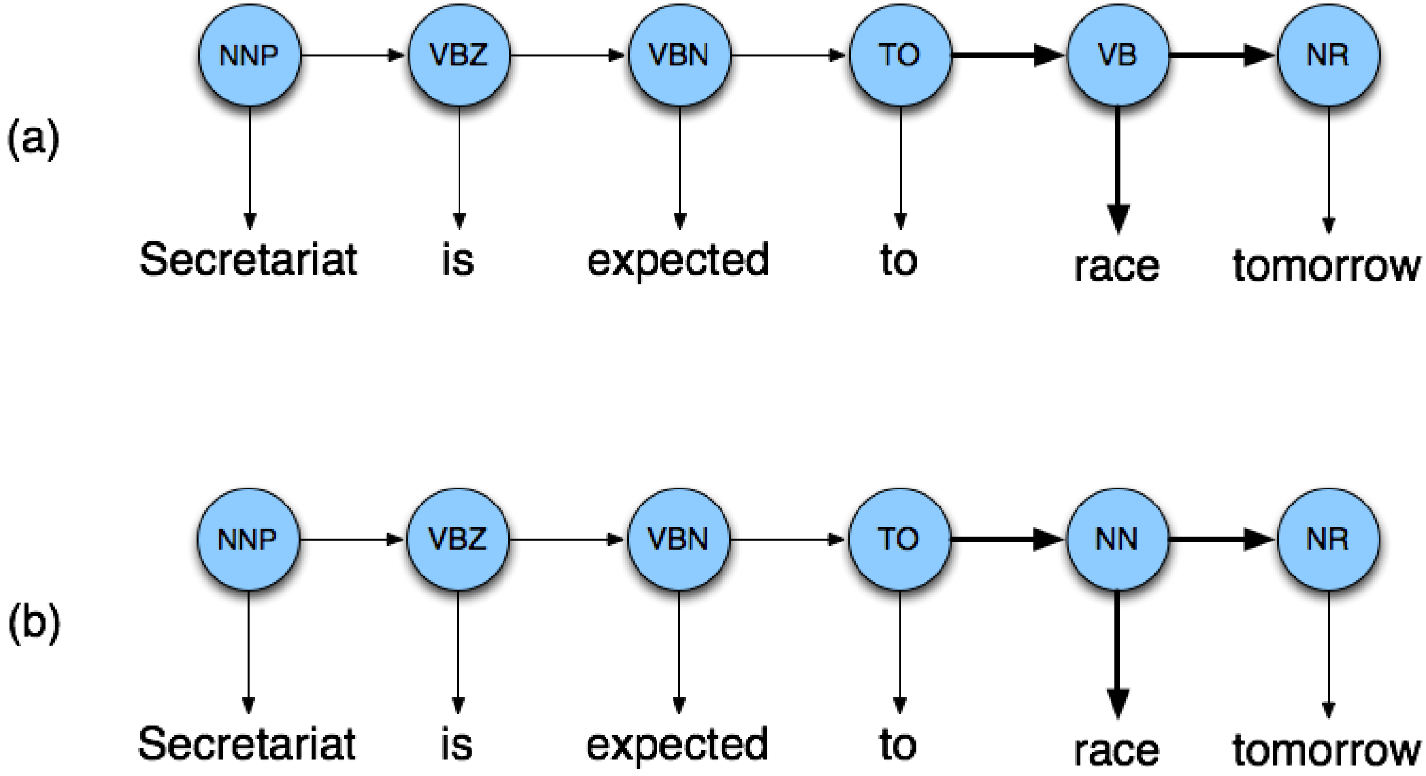
$$P_{GT}(w_i) = \frac{(c_i + 1)N_{c_i+1}}{N N_{c_i}}$$

- as if we used $c_i^* = (c_i + 1)\frac{N_{c_i+1}}{N_{c_i}}$ in MLE,
  discounting c_i and reallocating probability mass to unseen words.

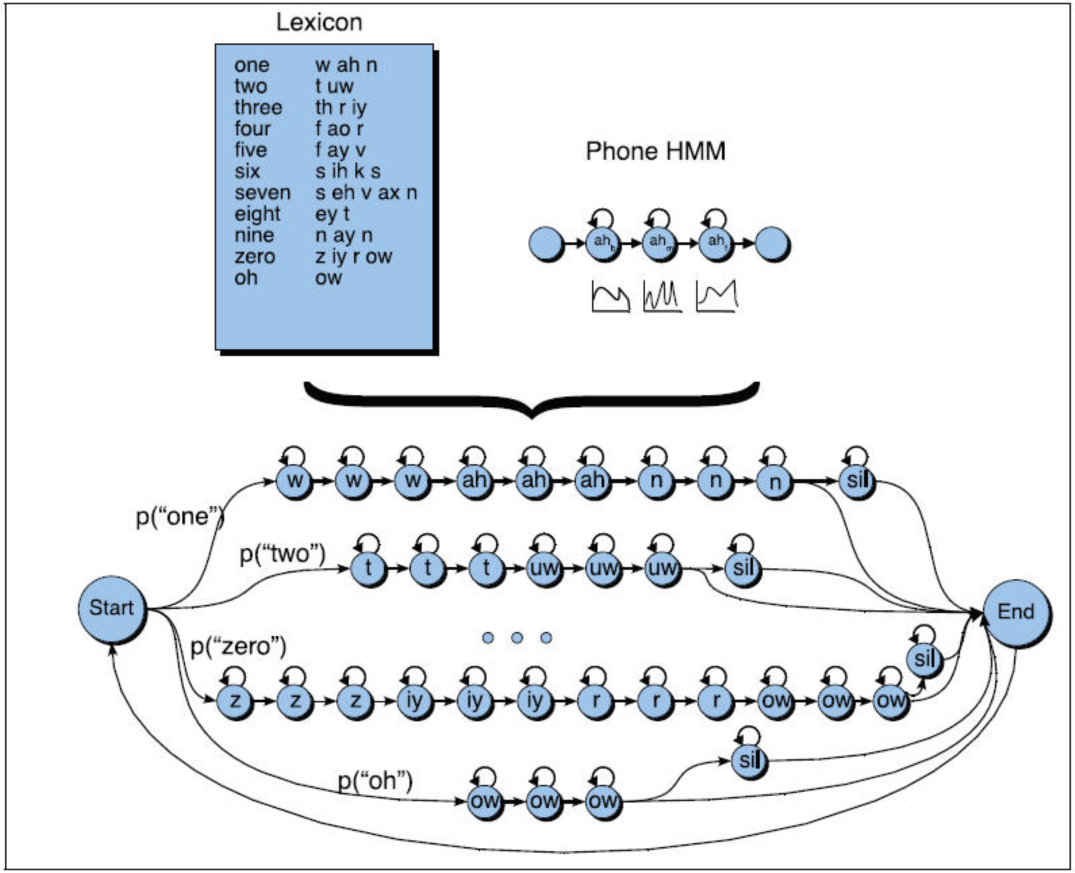
- Count the hapaxes $\rightarrow$ estimate the count of types unseen in training: $P_{GT}(\text{unseen}) = N_1/N$.

From n-grams models to Markov chains

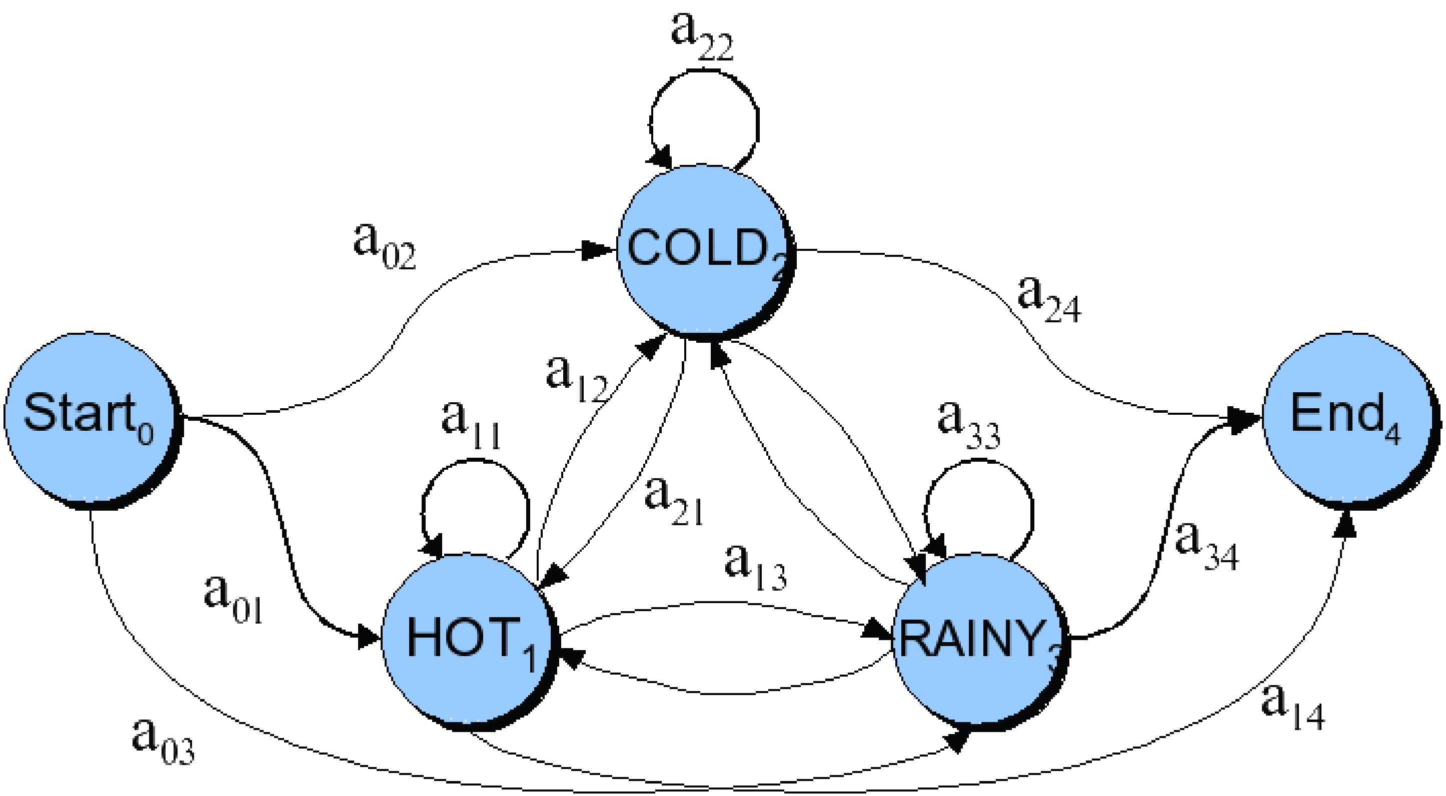
Part-of-Speech Tagging



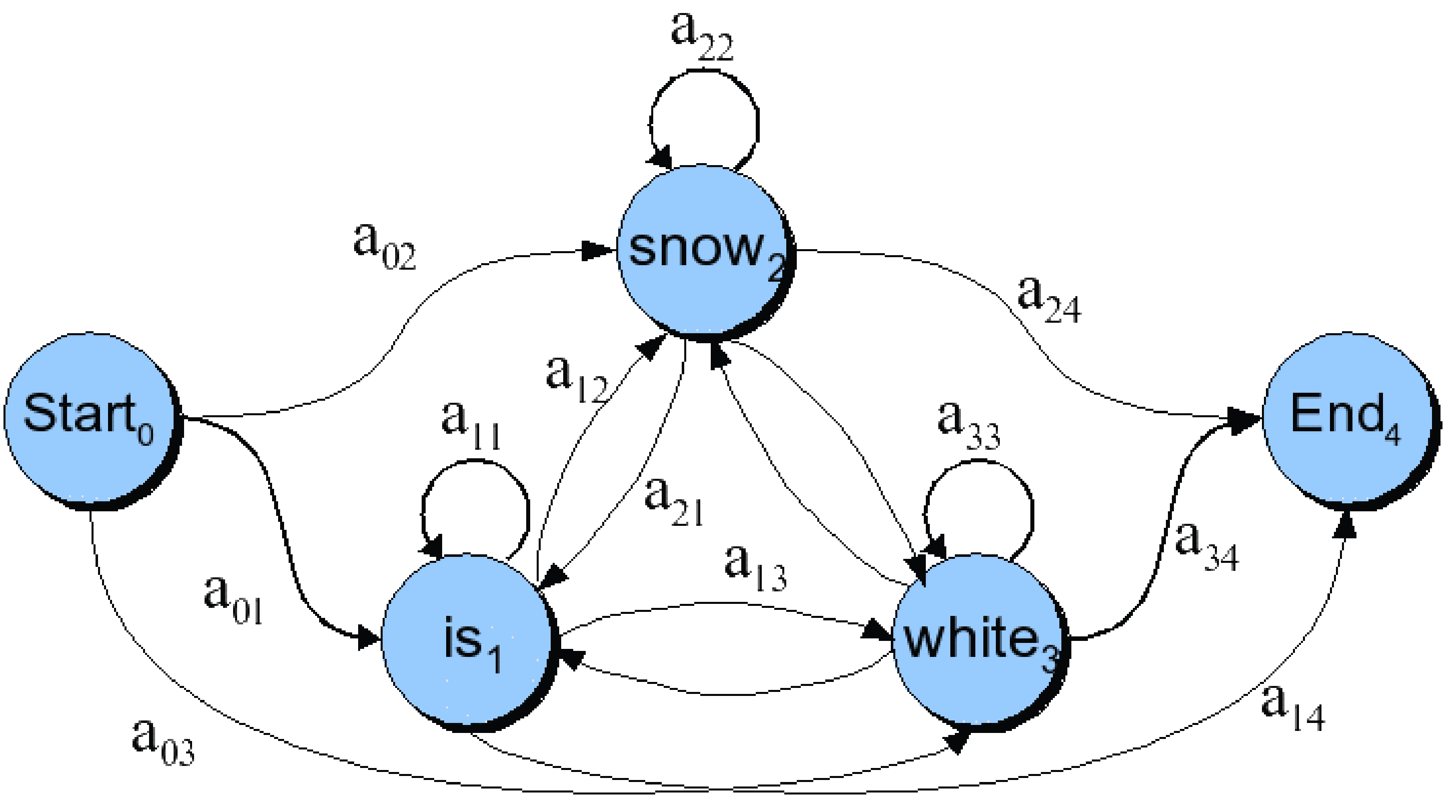
Speech recognition



Markov Chain for Weather



Markov Chain for Words



Markov Chain: "First-order observable Markov Model"

- Set of states Q. The state at time t is q_t.
- a_{ij}: probability transitioning $q_i \rightarrow q_j$.
- Transition matrix $A = (q_{ij})$.
- Current state depends **only** on previous state:

$$P(q_i|q_1 \dots q_{i-1}) = P(q_i|q_{i-1})$$

(Hidden) Markov Models

$Q = q_1 q_2 \ldots q_N$	a set of **states**
$A = a_{01} a_{02} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_N$	a set of **observations**, each one drawn from a vocabulary $V = v_1, v_2, ..., v_V$.
$B = b_i(o_t)$	a set of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state i
q_0, q_{end}	special **start and end states** that are not associated with observations

(Hidden) Markov Models

Next task:

learn how to estimate the parameters.

- Markov chain: use n-gram probabilities
- Markov models: decoding with Viterbi algorithm
- Hidden Markov models: parameter estimation with Expectation-Maximization Algorithm

See you on Thursday!