CS114 Lecture 23

Review

NOTE: These slides are just a reminder of the topics.

Use the course slides and the book for the details.

May 1, 2013

Professor Meteer

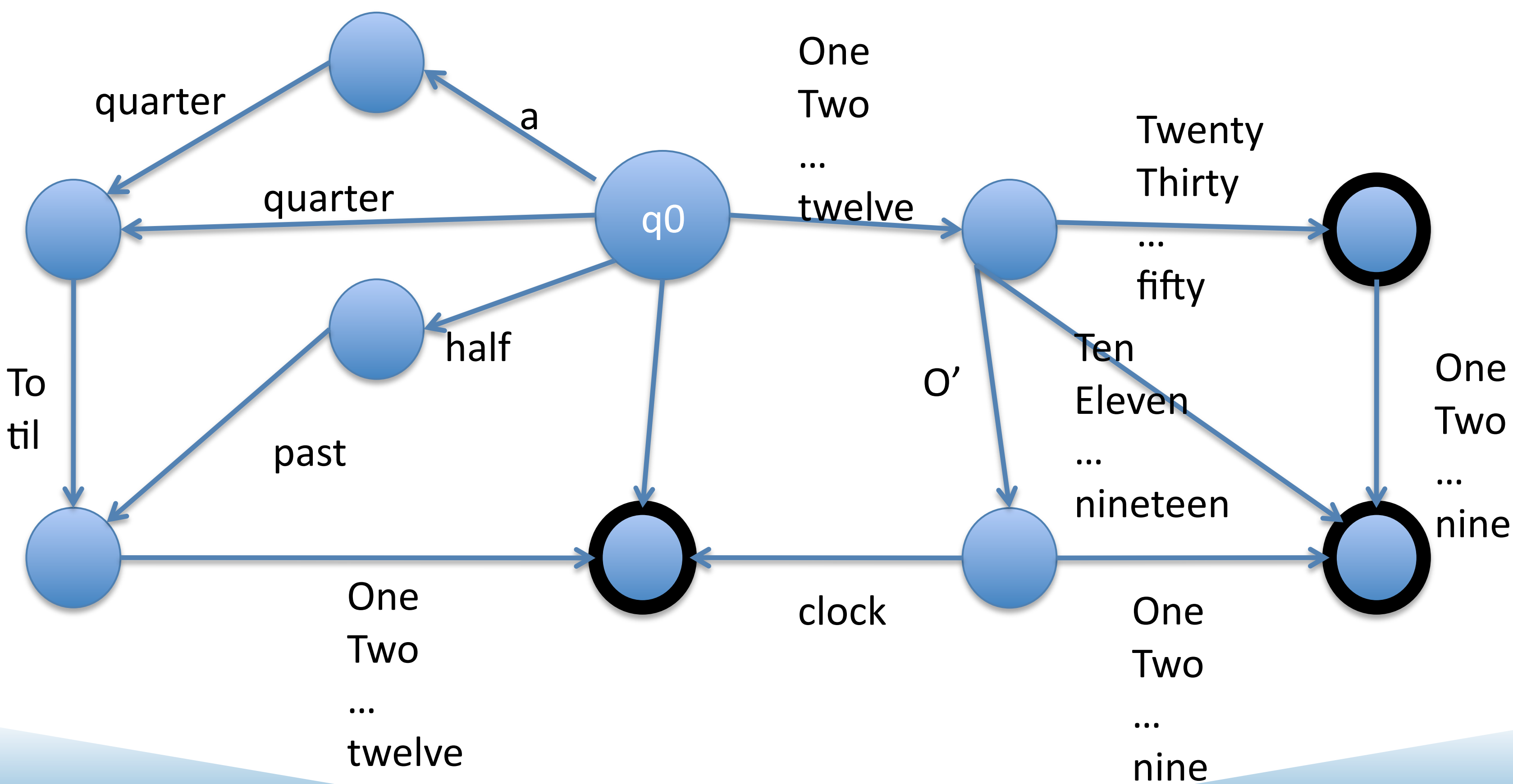
Review Part 1

- Linguistics: Morphology, POS
- Ambiguity
- FSAs
- Ngrams
  - What are some other applications? Spelling correction, text generation
- Viterbi algorithm and minimum distance
- Other applications of FSAs and HMMs

FSA's time of day

- Think about the data
  - One o'clock
  - Five twenty three
  - Quarter to nine
  - Six oh four
  - Half past twelve

FSAs: Time of day

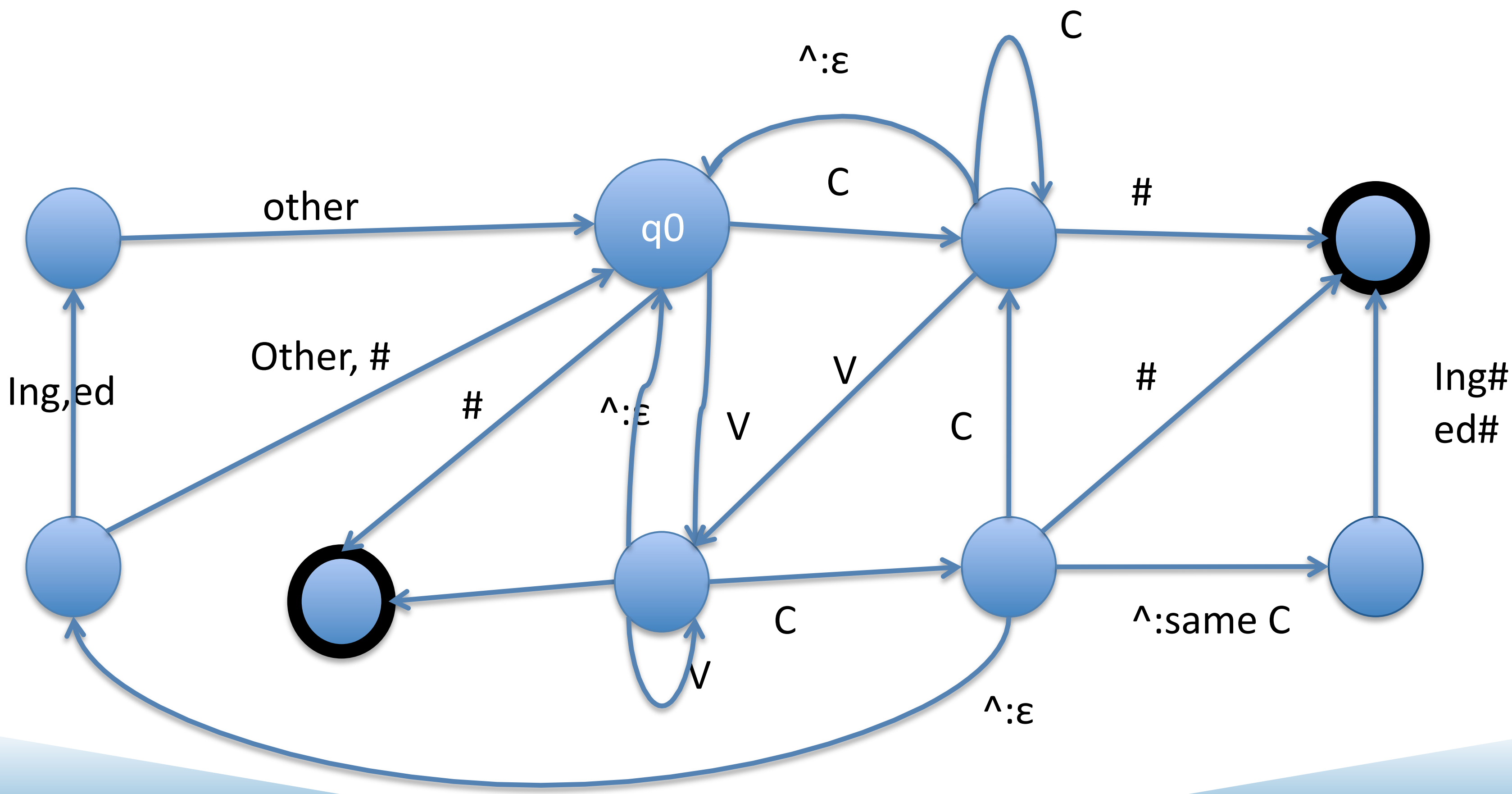


Doubling Consonants

- Look to the data
  - Tap, tapped, tapping, tape, taping
  - Bat, batted, batting, bate, bating

Transducer for doubling consonants

Tap^ing ..> Tapping



The Three Basic Problems for HMMs

Jack Ferguson at IDA in the 1960s

- Problem 1 (**Evaluation**):
  - Given the observation sequence $O=(o_1 o_2 \ldots o_T)$, and an HMM model $\Phi = (A,B)$, how do we efficiently compute $P(O|\Phi)$, the probability of the observation sequence, given the model
- Problem 2 (**Decoding**):
  - Given the observation sequence $O=(o_1 o_2 \ldots o_T)$, and an HMM model $\Phi = (A,B)$, how do we choose a corresponding state sequence $Q=(q_1 q_2 \ldots q_T)$ that is optimal in some sense (i.e., best explains the observations)
- Problem 3 (**Learning)**:
  - How do we adjust the model parameters $\Phi = (A,B)$ to maximize $P(O|\Phi)$?

Hidden Markov Models

- States Q = $q_1, q_2 \ldots q_{N;}$
- Observations O= $o_1, o_2 \ldots o_{N;}$
  - Each observation is a symbol from a vocabulary V = $\{v_1, v_2, \ldots v_V\}$
- Transition probabilities
  - Transition probability matrix A = $\{a_{ij}\}$

$$a_{ij} = P(q_t = j \mid q_{t-1} = i) \quad 1 \le i, j \le N$$

- Observation likelihoods
  - Output probability matrix B=$\{b_i(k)\}$

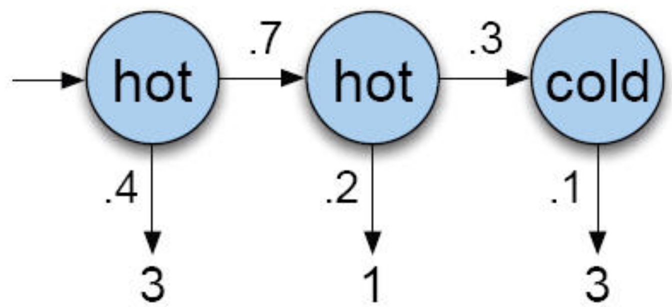
$$b_i(k) = P(X_t = o_k \mid q_t = i)$$

- Special initial probability vector π

$$\pi_i = P(q_1 = i) \quad 1 \le i \le N$$

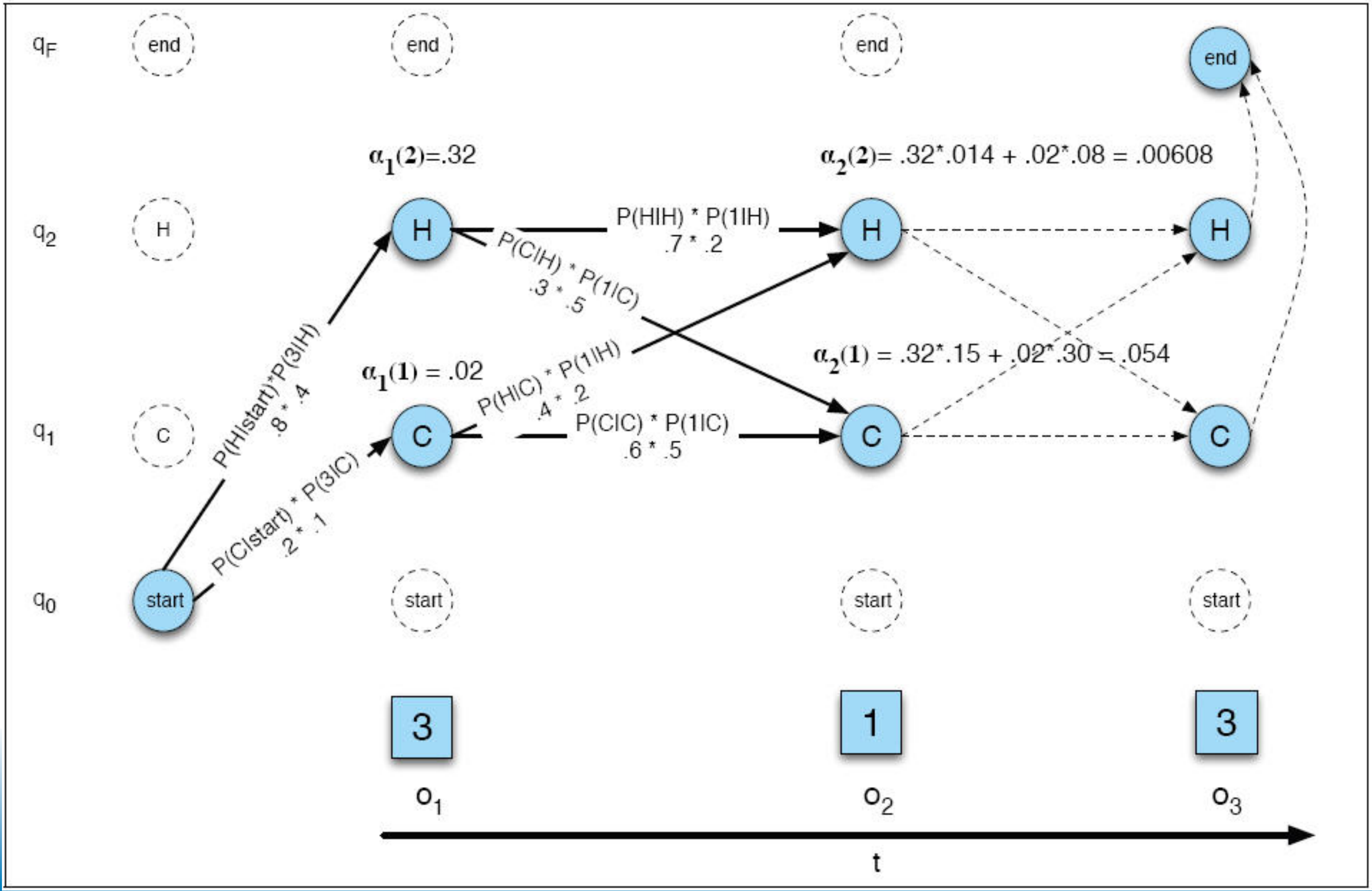
Joint probability

The computation of the joint probability of the ice cream events 3 – 1 – 3 and the hidden state sequence Hot Hot Cold



To find the most likely you would have to compute the probability for every sequence of hidden states. Too slow!

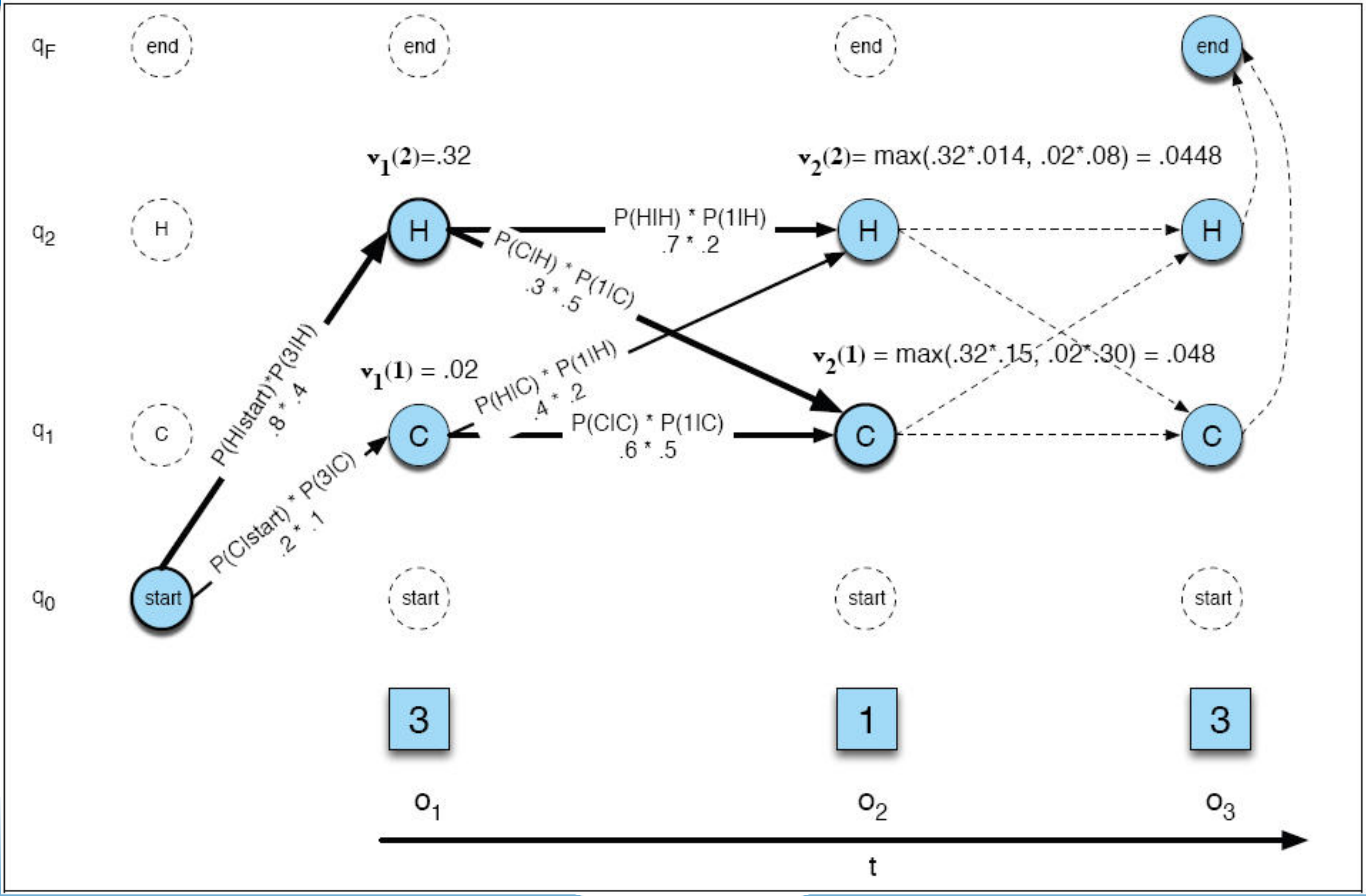
Dynamic Programming: Forward Algorithm



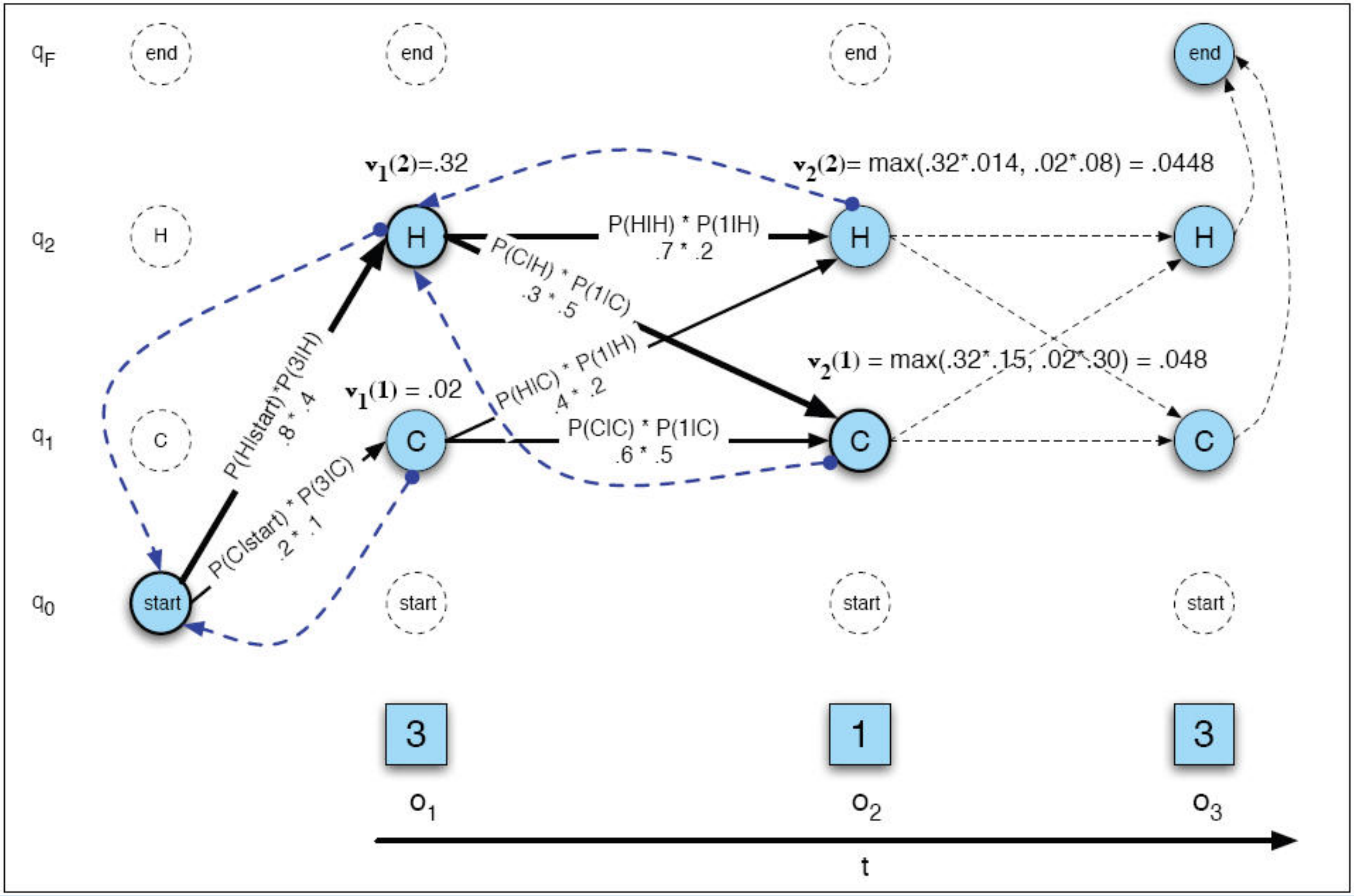
Viterbi Summary

- Create an array
  - With columns corresponding to inputs
  - Rows corresponding to possible states
- Sweep through the array in one pass filling the columns left to right using our transition probs and observations probs
- Dynamic programming key is that we need only store the MAX prob path to each cell, (not all paths).

Viterbi Trellis



Viterbi Trellis with Backtrace



Error Analysis: Confusion Matrix

-

	IN	JJ	NN	NNP	RB	VBD	VBN
IN	—	.2			.7		
JJ	.2	—	**3.3**	2.1	1.7	.2	**2.7**
NN		**8.7**	—				.2
NNP	.2	**3.3**	**4.1**	—	.2		
RB	**2.2**	2.0	.5		—		
VBD		.3	.5			—	**4.4**
VBN		**2.8**				**2.6**	—

- See what errors are causing problems
  - Noun (NN) vs ProperNoun (NNP) vs Adj (JJ)
  - Preterite (VBD) vs Participle (VBN) vs Adjective (JJ)

Semantics for a sentence

LIST FLIGHTS ORIGIN

Show me flights from Boston

DESTINATION DEPARTDATE

to San Francisco on Tuesday

DEPARTTIME

morning

HMMs for semantics

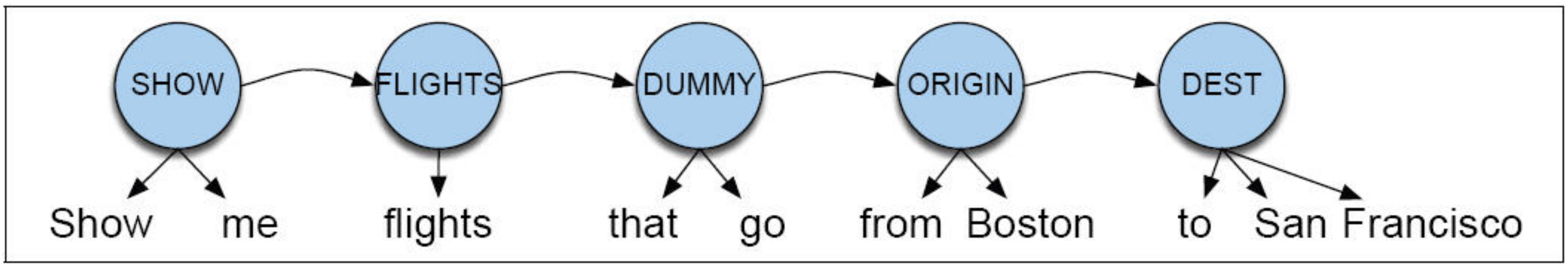
- Idea: use an HMM for semantics, just as we did for ASR (and part-of-speech tagging, etc)
- Hidden units:
  - Semantic slot names
    - Origin
    - Destination
    - Departure time
- Observations:
  - Word sequences

Hidden Markov Model Tagging

- Using an HMM to do POS tagging is a special case of *Bayesian inference*
  - Foundational work in computational linguistics
  - Bledsoe 1959: OCR
  - Mosteller and Wallace 1964: authorship identification
- It is also related to the "noisy channel" model that's the basis for ASR, OCR and MT

HMM model of semantics – Pieraccini et al (1991)

- Input is the set of words
- Output is the set of semantic states



Good-Turing

- Notation: N_x is the frequency-of-frequency-x
  - So $N_{10}=1$
    - Number of fish species seen 10 times is 1 (carp)
  - $N_1=3$
    - Number of fish species seen 1 is 3 (trout, salmon, eel)
- To estimate total number of unseen species
  - Use number of species (words) we've seen once
  - $c_0^* = c_1$ $\quad p_0 = N_1/N$

$$c^* = (c+1)\frac{N_{c+1}}{N_c}$$

- All other estimates are adjusted (down) to give probabilities for unseen

Slide from Josh Goodman

Good-Turing Intuition

- Notation: N_x is the frequency-of-frequency-x
  - So N_{10}=1, N_1=3, etc
- To estimate total number of unseen species
  - Use number of species (words) we've seen once
  - c_0^* =c_1 $p_0 = N_1/N$ p_0=N_1/N=3/18

$$P^*_{GT}(\text{things with frequency zero in training}) = \frac{N_1}{N}$$

- All other estimates are adjusted (down) to give probabilities for unseen

$$c^* = (c+1)\frac{N_{c+1}}{N_c}$$

P(eel) = c*(1) = (1+1) 1/ 3 = 2/3

Slide from Josh Goodman

Could just spread 1s over 0s

Carp	10	10
Perch	3	3
WF	2	2
Trout	1	1
Salmon	1	1
Eel	1	1
Catfish	0	1
Bass	0	1
TOTAL	18	

- Prob of things that occurred once
- 1\18 + 1\18 + 1\18 = 3\18
- Add one to zero counts
- Spread probability over 1s and 0s
- 3/18 / 5 = .066

GT Fish Example

- OR use the 1s for 0s (3/18 spread over2 species)
- AND Look at the things that happened 2s to share with 1s
  - C(whitefish) = 2 happened once
  - Discount 1s by 2/3
- LOTS OF ALTERNATIVES! Just estimates

	unseen (bass or catfish)	trout
c	0	1
MLE p	$p = \frac{0}{18} = 0$	$\frac{1}{18}$
c^*		$c^*(\text{trout}) = 2 \times \frac{N_2}{N_1} = 2 \times \frac{1}{3} = .67$
GT p^*_{GT}	$p^*_{\text{GT}}(\text{unseen}) = \frac{N_1}{N} = \frac{3}{18} = .17$	$p^*_{\text{GT}}(\text{trout}) = \frac{.67}{18} = \frac{1}{27} = .037$

Review

- Major topics for this section
  - Syntax
  - Parsing
  - Semantics
  - Lexical Semantics
- Use the slides to indicate what's important and the book to describe it in more detail
  - If it's in the book and not in the slides it won't be on the test
  - but slides are bullet points and picture—use the book to know how to talk about these points

Syntax

- Know your basic phrase types
  - VP does not mean Vice President
- Terms to know
  - Derivation
  - Overgenerate
  - Syntactic grammars
  - Dependency grammars
  - Verb subcategorization

Parsing Types

- CFGS
  - Top down, bottom up
  - CKY
  - Earley's algorithm
- Probabalistic CFGs
- Unification Grammars
- Chunking
- Partial parsing

Semantics

- Synonyms vs. Similar vs. Related
- Wordnet
- Word Sense Disambiguation
  - Feature vectors
  - Collocational vs. bag of words
- Similarity metrics
  - Thesaurus-based vs. distributional
  - Context vectors
- Entropy and Mutual Information

Weighting: Mutual Information

- **Mutual information:** between 2 random variables X and Y

$$I(X,Y) = \sum_x \sum_y P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)}$$

- **Pointwise mutual information**: measure of how often two events x and y occur, compared with what we would expect if they were independent:

$$I(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

Mutual information intuition

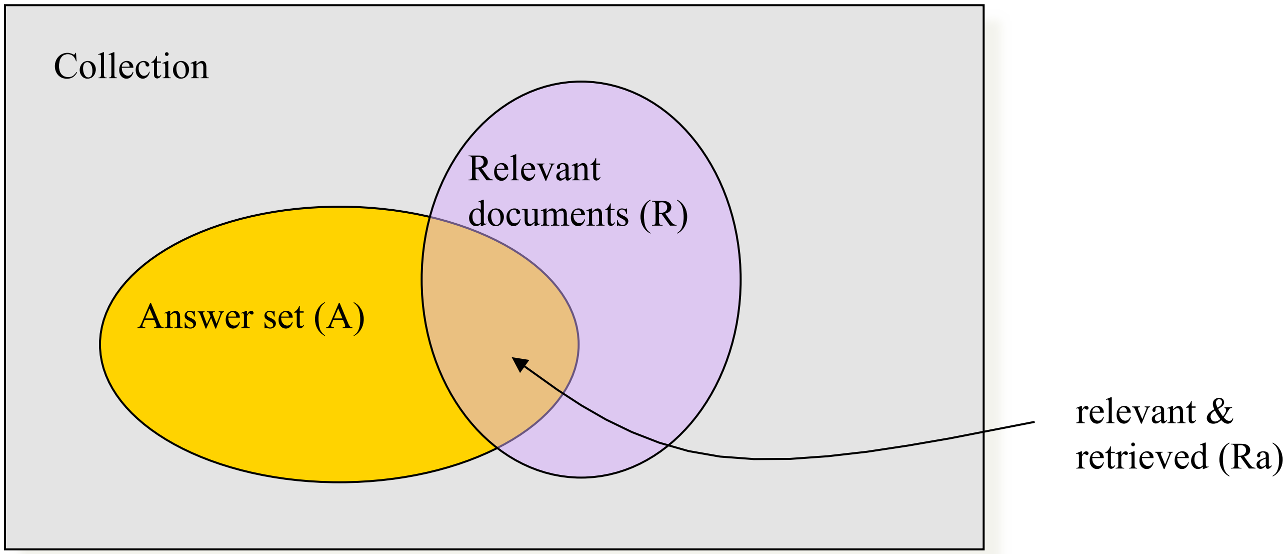
- Objects of the verb *drink*

Object	Count	PMI assoc	Object	Count	PMI assoc
bunch beer	2	12.34	wine	2	9.34
tea	2	11.75	water	7	7.65
Pepsi	2	11.75	anything	3	5.15
champagne	4	11.75	much	3	5.15
liquid	2	10.53	it	3	1.25
beer	5	10.20	<SOME AMOUNT>	2	1.22

Evaluation

- Precision and recall
- Intrinsic and extrinsic
- Inter-annotator agreement

Classic IR Terminology



- Recall is the fraction of the relevant documents which has been retrieved
  Recall = |Ra| / |R|
- Precision is the fraction of the retrieved documents which is relevant
  Precision = |Ra| / |A|

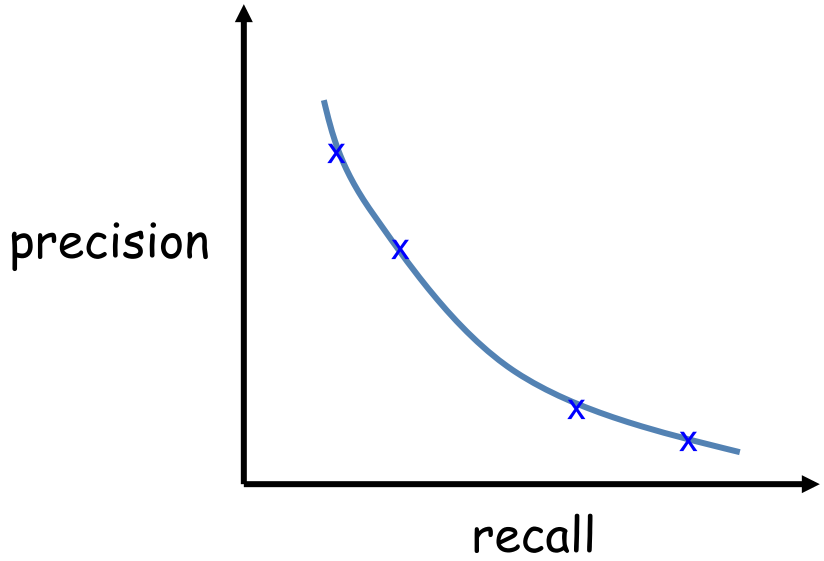
Evalutation Metrics from IR

- Precision = number of relevant items retrieved
  - number of items retrieved
- Recall = number of relevant items retrieved
  - number of relevant items in collection

- Aim to maximize both, but compromises are needed.
- Relevance is highly subjective
  - doesn't allow for "quite relevant", "not very .."
  - assesses relevance of a doc. to query put to system, not to the information need the user has.

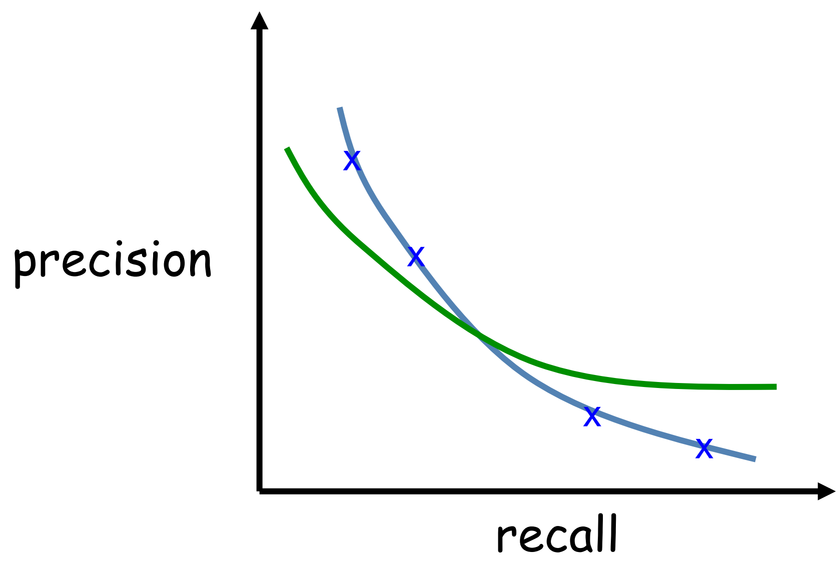
Precision/ Recall Curves

- There is a tradeoff between Precision and Recall
- So measure Precision at different levels of Recall



Precision/ Recall Curves (Cont.)

- Difficult to determine which of these two hypothetical results is better:



F-Measure: The Harmonic Mean

- The harmonic mean combines Recall & Precision into a single number ranging from 0 to 1:

$$F(j) = \frac{2}{\dfrac{1}{r(j)} + \dfrac{1}{P(j)}}$$

P(j) - precision of *j*-th document in ranking;
r(j) – recall of *j*-th document in ranking;

- If F(j) = 0 – no relevant docs have been retrieved;
- If F(j) = 1 – all ranked docs are relevant;
- The harmonic mean assumes *high* value only when both recall & precision are *high*.

Top Level

- Lexical Semantics, word sense disambiguation
- Corpus analysis and annotation
- Discourse: Coreference
- Features
- Classifiers
- Discourse Structure

Discourse - Coreference

- Coreference
  - Kinds of reference phenomena
  - Constraints on co-reference
  - Anaphora Resolution
    - Hobbs
    - Loglinear
  - Coreference

Some terminology

- Reference: Process by which speakers use words Victoria Chen and she to denote a particular person
  - **Referring expression**: Victoria Chen, she
  - **Referent**: the actual entity (but as a shorthand we might call "Victoria Chen" the referent).
  - Victoria Chen and she "**corefer**"
  - **Antecedent**: Victoria Chen
  - **Anaphor**: she

Coreference Example

- **Victoria Chen**, **Chief Financial Officer** of **Megabucks Banking Corp** since 2004, saw **her** pay jump 20%, to $1.3 million, as **the 37-year-old** also became **the Denver-based financial-service company's president**. It has been ten years since **she** came to Megabucks from rival Lotsabucks.

Coreference resolution

- **Victoria Chen**, **Chief Financial Officer of Megabucks Banking Corp** since 2004, saw **her** pay jump 20%, to $1.3 million, as **the 37-year-old** also became **the Denver-based financial-service company's president**. It has been ten years since **she** came to Megabucks from rival Lotsabucks.
  - {Victoria Chen, Chief Financial Officer of Megabucks Banking Corp, her, the 37-year-old, the Denver-based financial-services company's president, she}
  - {Megabucks Banking Corp., the Denver-based financial-services company, Megabucks}
  - {her pay}
  - {Lotsabucks}

A loglinear model

- Supervised machine learning
- Train on a corpus in which each pronoun is labeled with the correct antecedent
- In order to train: We need to extract
  - Positive examples of referent-pronoun pairs
  - Negative example of referent-pronoun pairs
  - Feature for each one
- Then we train model to predict 1 for true antecedent and 0 for wrong antecedents

Features

- Strict gender (T/F)
  - e.g. male pronoun Pro_i with male antecedent NP_j
- Compatible gender (T/F)
  - e.g. male pronoun Pro_i with antecedent NP_j of unknown gender
- Strict number (T/F)
  - e.g. singular pronoun with singular antecedent
- Compatible number (T/F)
  - e.g. singular pronoun with antecedent of unknown number

Features

- Strict gender (T/F)
  - e.g. male pronoun Pro_i with male antecedent NP_j
- Compatible gender (T/F)
  - e.g. male pronoun Pro_i with antecedent NP_j of unknown gender
- Strict number (T/F)
  - e.g. singular pronoun with singular antecedent
- Compatible number (T/F)
  - e.g. singular pronoun with antecedent of unknown number

Features

- Machine learning paradigm
  - Target and features
- Applications
  - POS tagging, parsing, speech recognition
  - Word sense disambiguation
  - Semantic role labeling
- Types of features
  - Boolean, multivaried

KeyWords Detector: tf-idf

- The tf–idf weight (term frequency–inverse document frequency) is a a statistical measure used to evaluate how important a word is to a document in a collection or corpus.
- The importance increases proportionally to the number of times a word appears in the document (*term frequency*) but is offset by the frequency of the word in the corpus (*inverse document frequency*).
- We are using tf-idf score as a main tool for keywords detection
  - For example, word “time” has a very high document frequency (df), which converts to a low idf count and overall low tf-idf score of this word
  - On the other hand, multiword “bubba_watson” has much lower df, and, correspondingly, higher idf and tf-idf
- It’s a very good technique, but it can produce lousy keywords in two cases:
  - it never (or rarely) seen a word before, like “twiloightandtheb”
  - there are no interesting words in the document

Topic Model Example:

central => |General_English:0.0195193|

bank => |commercials:0.317051|

central_bank => |business_news:0.93075|

pullback => |business_news:0.830826|world_news: 0.58002|

home => |General_English:0.0234851|

depot => |business_news:0.829389|

home_depot => |business_news:0.958285|weather: 0.305589|

critic => |political_news:0.326691|world_news: 0.0618789|

Corpus analysis and annotation

- LDC: Treebank, etc.
- Corpus creation process
  - Defining guidelines
  - Training and test
- Corpus evaluation
  - Inter-annotator agreement
  - Precision and recall

Discourse Structure

- Discourse Structure
  - Textiling
- Cohesion
- Coherence
  - Hobbs coherence relations
  - Rhetorical Structure Theory