AT&T Labs-Research

# Adaptive Learning: From Supervised to Active Learning of Statistical Models for Natural Language and Speech Processing

**Giuseppe Riccardi***
**Dilek Hakkani-Tür^**
**Gokhan Tur♠**

* Currently with University of Trento, Italy
^ Currently with ICSI,USA
♠ Currently with SRI, USA

Eurospeech 2003, Geneva

# Acknowledgements

Mazin Rahim
Robert Schapire
Narendra Gupta
Jerry Wright

# Outline

- Learning Dimension:
  - Passive vs. Active Learning
  - Supervised vs Unsupervised Learning
  - Combining Active and Unsupervised Learning
- Application Dimension:
  - Classification (Text categorization, Part of Speech Tagging, Call Classification,…)
  - Automatic Speech Recognition
  - Syntactic Parsing

# Learning

- **Describe (natural) phenomenon**
  - *Apple falling off the tree (XVII century)*
  - *NASDAQ (XX century)*
- **Data collection (Experiment)**
  - **Experiments *vs* Measurements**

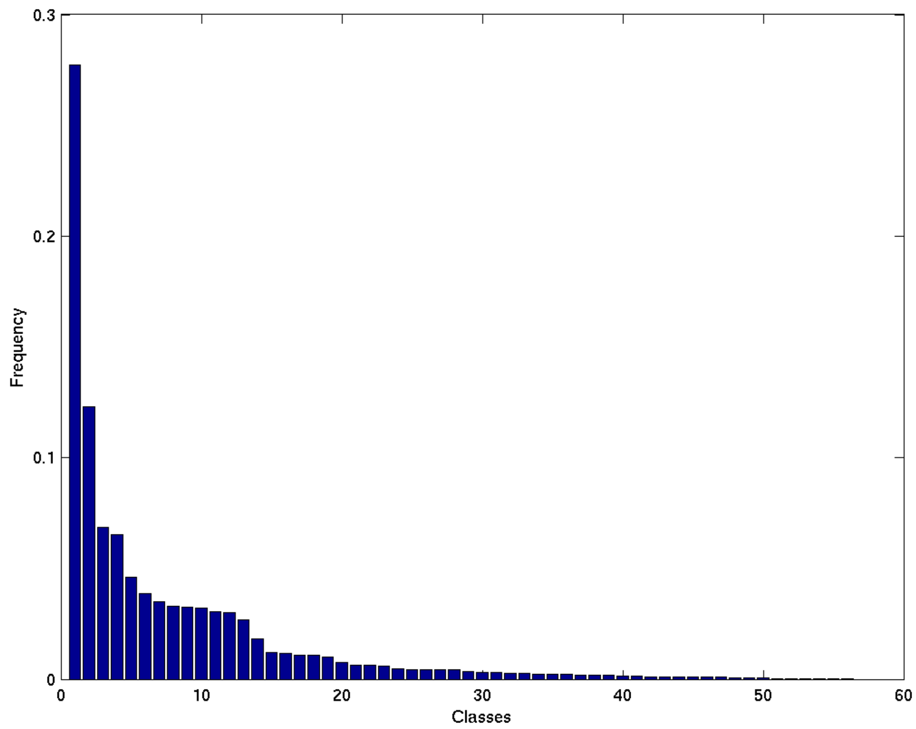
  "Do you like candidate X?"

  "Do you like candidate X or rather Y?"

- **Modeling data (Prediction)**
  - *What if I jump off a tree?*
  - *Is candidate Y going to win the election?*

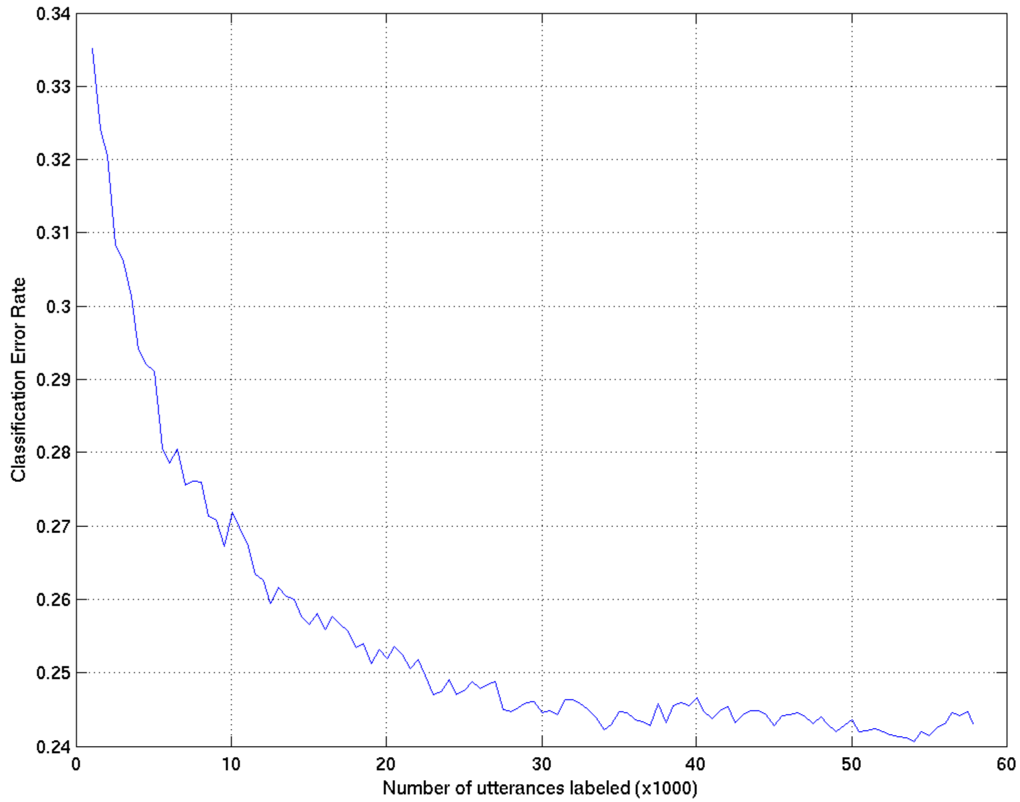
# Passive Learning

- Typical Class Distribution
  - Zipf's Law: *Frequency x Rank = Constant*
  - *Sample infrequent examples (tail of the distribution)*



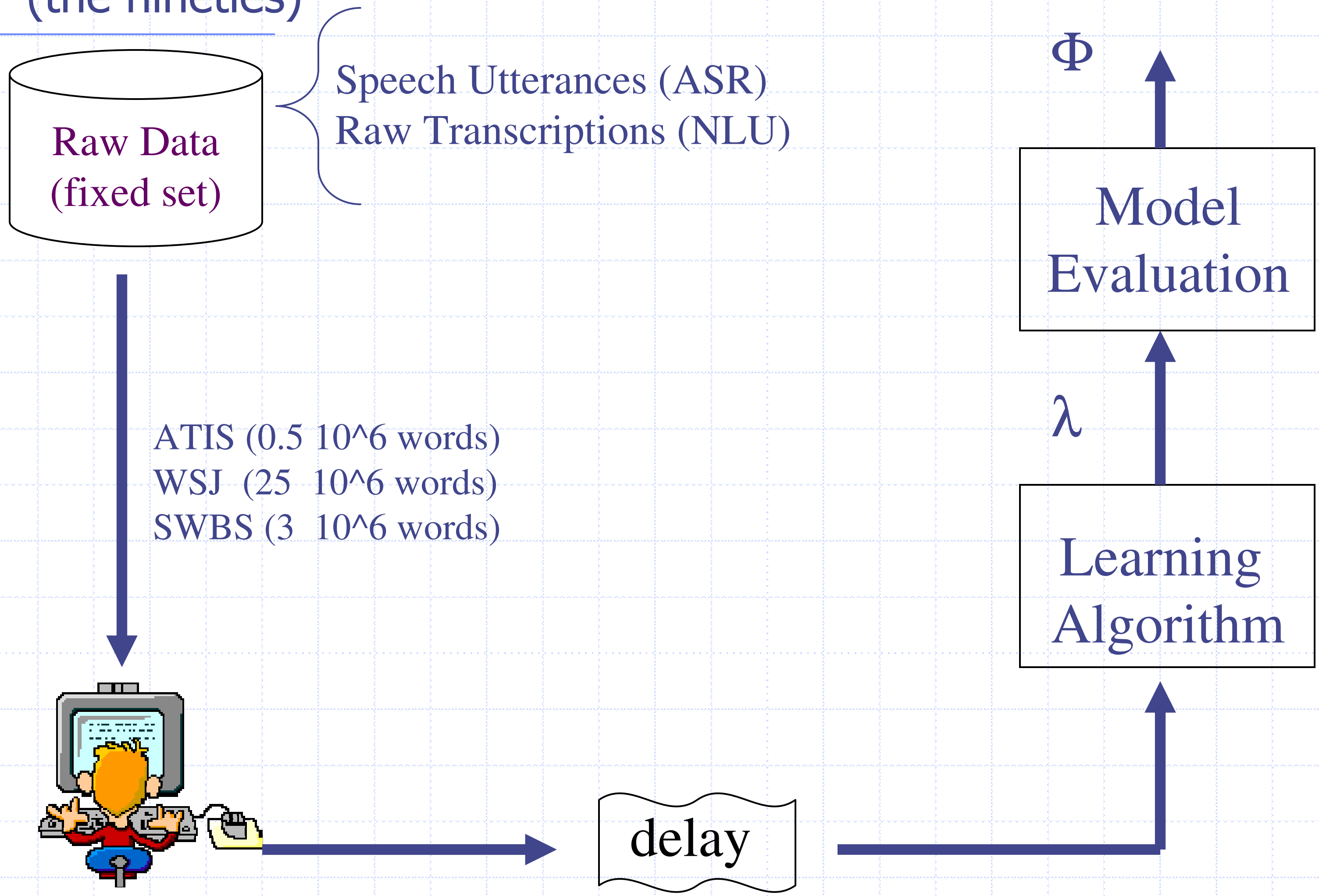
# Passive Learning

- Typical Learning Curve
  - "no data like more data"

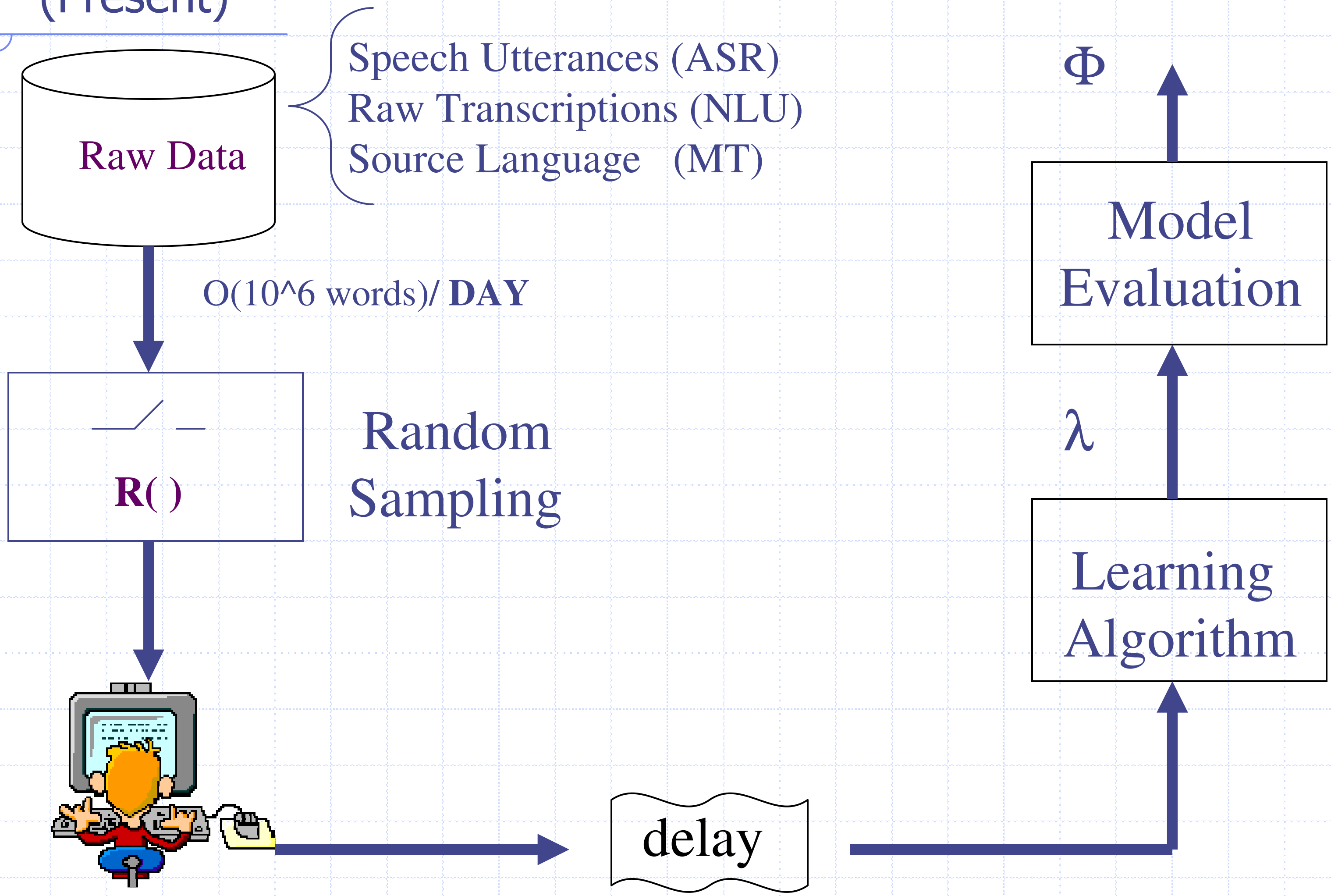


# Supervised Learning

(the nineties)



# Supervised Learning
(Present)



# Data Driven Learning

- The Eighties: (almost) no data, prior knowledge
- The Nineties: Data Driven Models
  - DARPA projects (ATIS, WSJ)
  - "no data like more data"
- Third Millenium
  - Terabytes of Data ("*Data Divide between University and Private Research*")
- Supervised Learning (*learning from examples*)
  - Small data set
  - Human intervention (labeling or annotation)
  - Delayed Response

# Maximum Likelihood (1)

- The General setting
- Data Samples (Measurements) i.i.d.
  - $X = \{x_1, \ldots, x_N\}$
- Underlying probability law p(X) with parameters $\boldsymbol{\theta}$
- $P(X|\boldsymbol{\theta}) = \prod_k p(x_k|\boldsymbol{\theta})$
  - (log) Likelihood function

# Maximum Likelihood (2)

- Example: Binary random variable

$X = \{x_1, x_1 \cdots, x_N\}$ Training set of data samples

$L(X,\theta) = P(X \mid \theta)$ Likelihood Function

$$\log L(X,\theta) = \log(p^{N_1}(1-p)^{N_2}) = N_1 \log p + N_2 \log(1-p)$$

$$\frac{d \log L(X,\theta)}{d\theta} = 0$$ Likelihood Maximization

$$p = \frac{N_1}{N_1 + N_2}$$

# Maximum Likelihood (3)

- Example: Language Modeling

$$
\begin{aligned}
P(W) &= P(w_1 w_2 \cdots w_N) \\
&= \prod_i P(w_i \mid w_1 \cdots w_{i-1}) \\
&= \prod_i P(w_i \mid w_{i-n+1} \cdots w_{i-1})
\end{aligned}
$$

# Example: Language Modeling

## Data Sparseness Problem

- Large Vocabulary (|V| ~ 50K)
- Generalization
  - I would like {a, to, the, this,..}
- Zipf's Law (frequency of n-gram ∝ 1/n)

## Maximum Likelihood (ML) Probability

$$P(w_i \mid w_{i-n+1},..,w_{i-1}) = \# w_1 w_2 \ldots w_i / \# w_1 w_2 \ldots w_{i-1}$$

## Discounted ML Probability

$$\hat{P}(w_i \mid w_{i-n+1},..,w_{i-1}) = \alpha(w_1 w_2 \ldots w_i) P(w_i \mid w_{i-n+1},..,w_{i-1})$$

# Discriminative Training

- The goal of ASR is to minimize the probability of error. This does not necessarily imply maximizing $P(X \mid \Phi)$.
- Discriminative Training methods are applied to maximize a function that provides better discrimination between classes.
- Automatic Speech Recognition
- Text Classification

# Adaptive Learning

- **Describe (natural) phenomenon**
  - *NASDAQ (*Measurements over a month in April)
  - $X = X_1, X_2, X_3, \ldots, X_N$
  - What if a war is going on?
  - $X = X_1(t), X_2(t), X_3(t), \ldots, X_N(t)$
  - Time dependent statistics
    - Stationary (e.g. seasonal effects)
    - Bursty (e.g. unforeseen events)
- **Adaptive Learning**
  - Prediction is based on **current estimates (input) and adapts (output).**
  - **State of the system**

# Adaptive Learning

- Definition
  - Adapt fast to changes in feature statistics
  - Learn new events
  - Minimize supervision
- Instead of assuming a fixed and given training data as in the passive learning, the data is dynamic and determined by the learner itself.

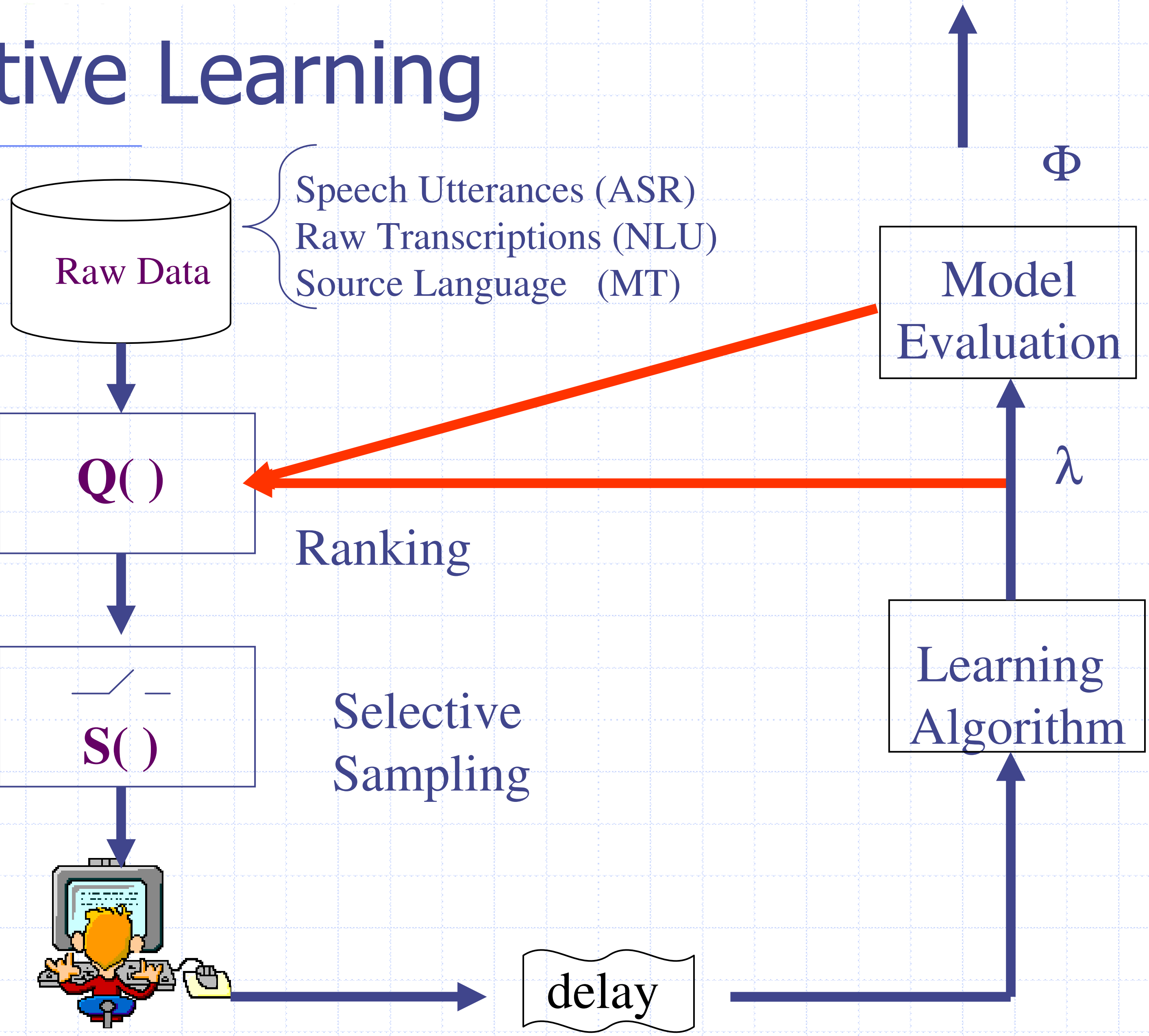
# Adaptive Learning

- Methods for adaptive learning:
  - Active learning
  - Unsupervised learning
  - Combining active and unsupervised learning

# Outline

- Algorithm Dimension:
  - Passive vs. Adaptive Learning
  - Active Learning
    - Certainty-based
    - Committee-based
  - Unsupervised Learning
  - Combining Active and Unsupervised Learning

# Active Learning



# Active Learning
(static)

- Sample space T is very large and finite (size N)

*Select* $K_{min}$ examples from T to label such that $\Delta\Phi$ is maximized on a random test set

- The number of combinations of k examples is very large (N!/k!(N-k)!)
- The number of permutations of k examples is very large (k!)

# Active Learning
(dynamic)

- Sample space $\mathrm{T}$ is very large (size N)
- At time t there are K(t) samples available

At time t, for a given K(t) in T,

*Compute* $K_{min}$ examples from K(t) to label such that $\Delta\Phi$ is maximized on a random test set

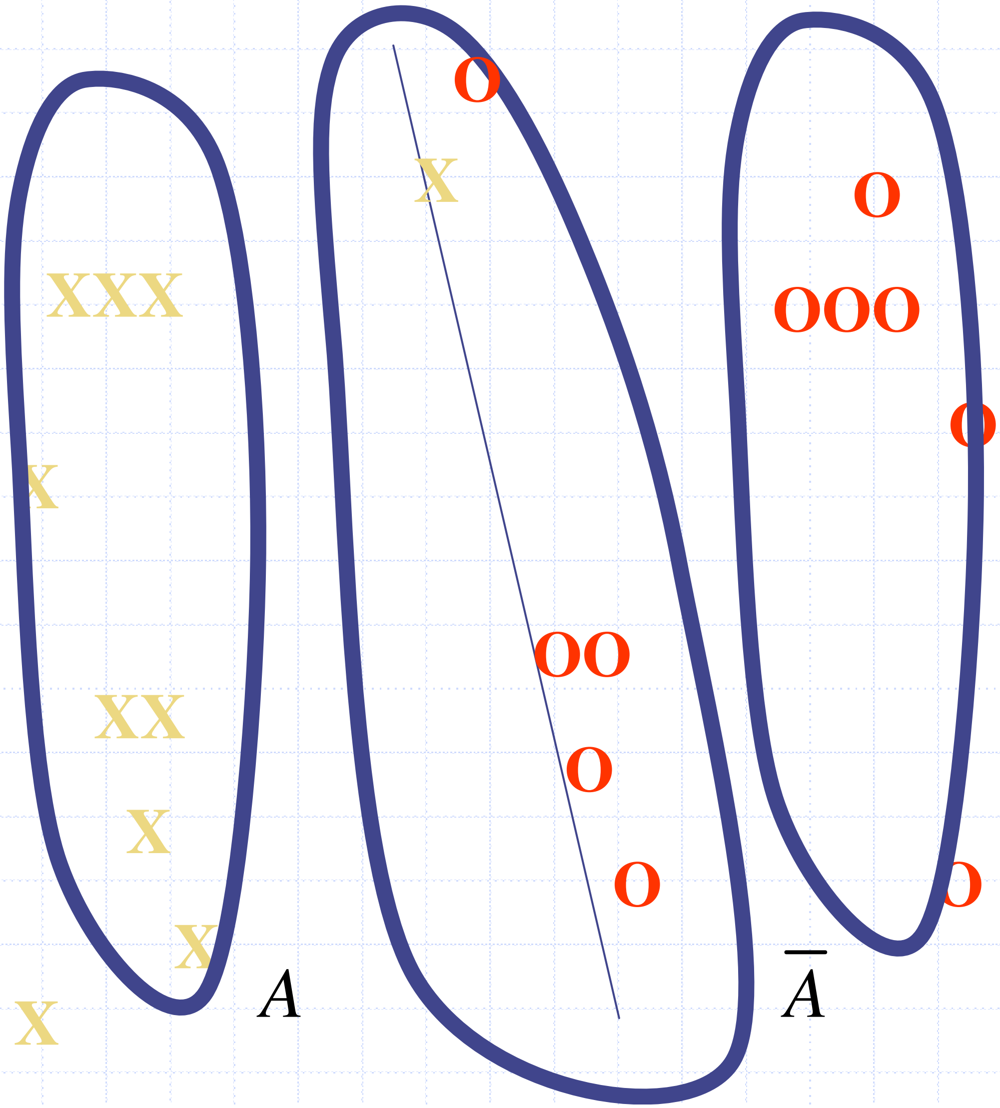
- Compute $\rightarrow$ Select from a given $\mathrm{T}$
- $t=\infty$

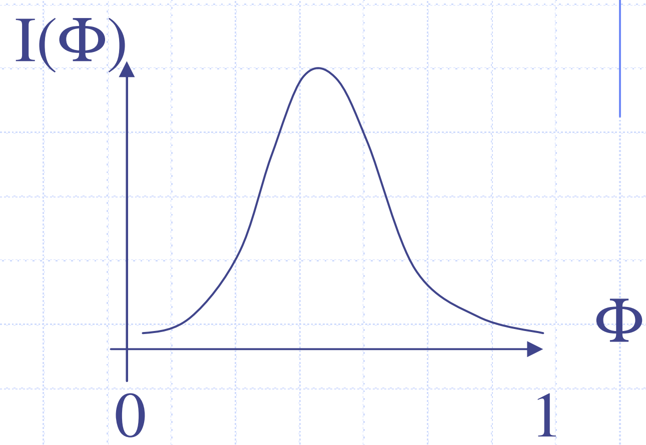
# Ranking Sample Space (1)

- $T = \{u_i\}$
  - Set of all examples
- $Q(u_i)=j$
  - Compute confidence scores for each example
    - Probability that example $u_i$ is correctly labeled by the current model $\lambda$
  - Sort
- Selective Sampling S()
  - $S(T)=(1,\ldots K_{min})$
- Label S(T)

# Ranking Sample Space (2)

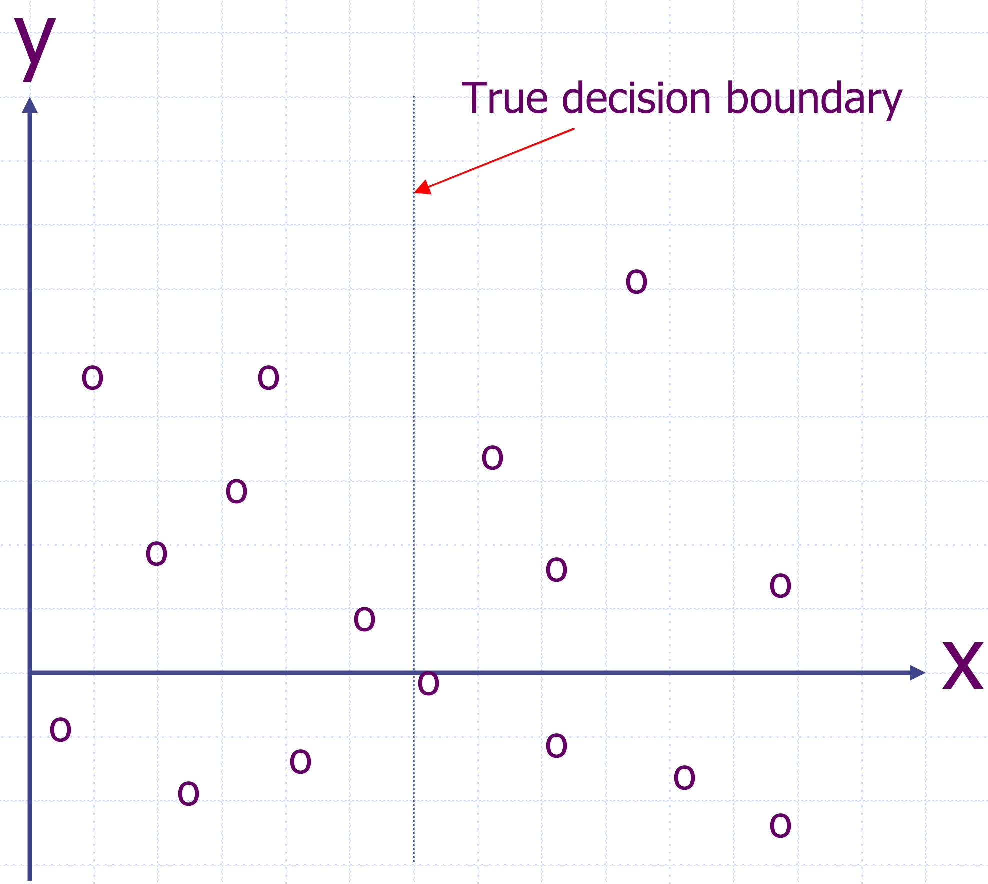
(classification case)



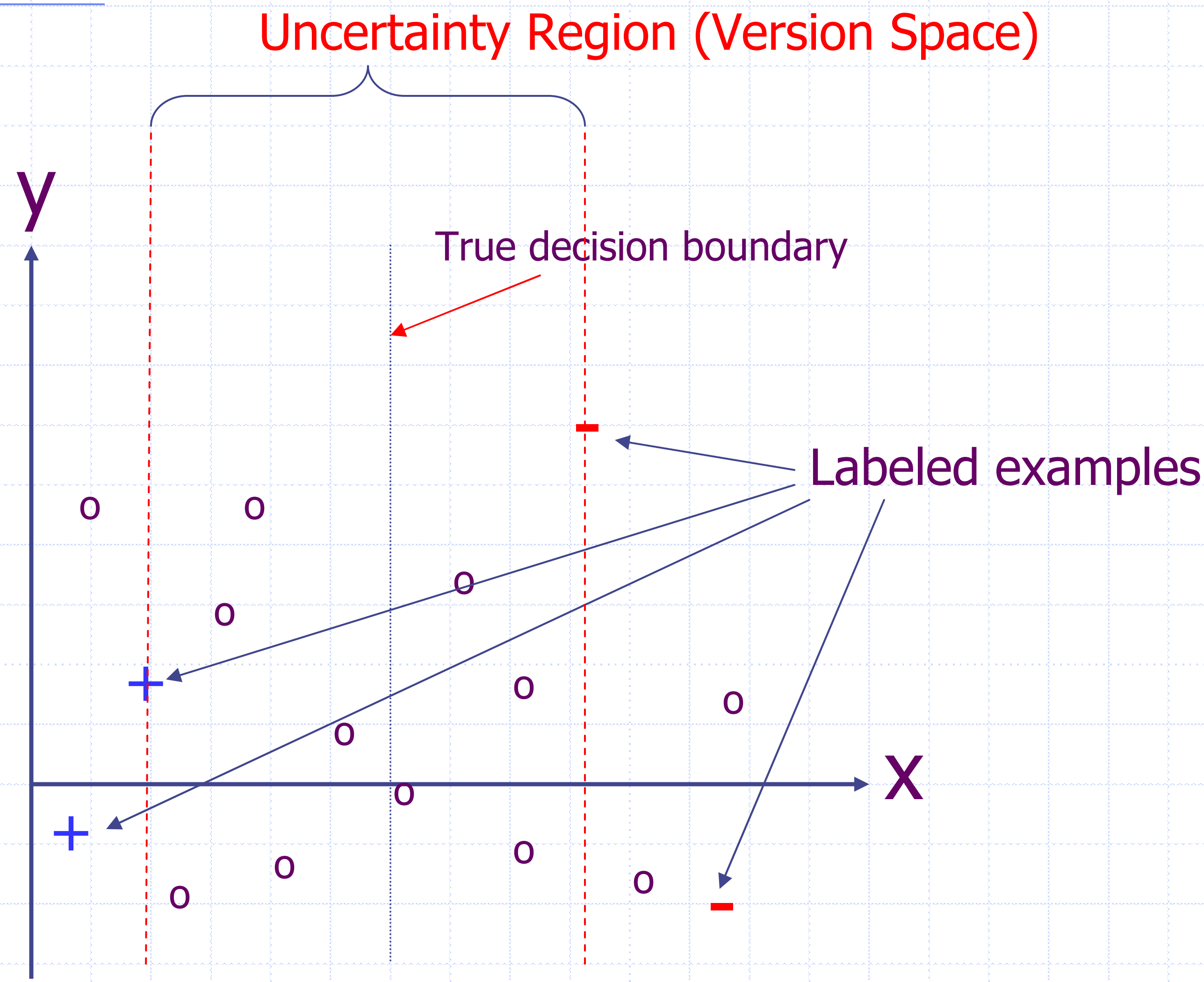



# A Simple Binary Classification Example

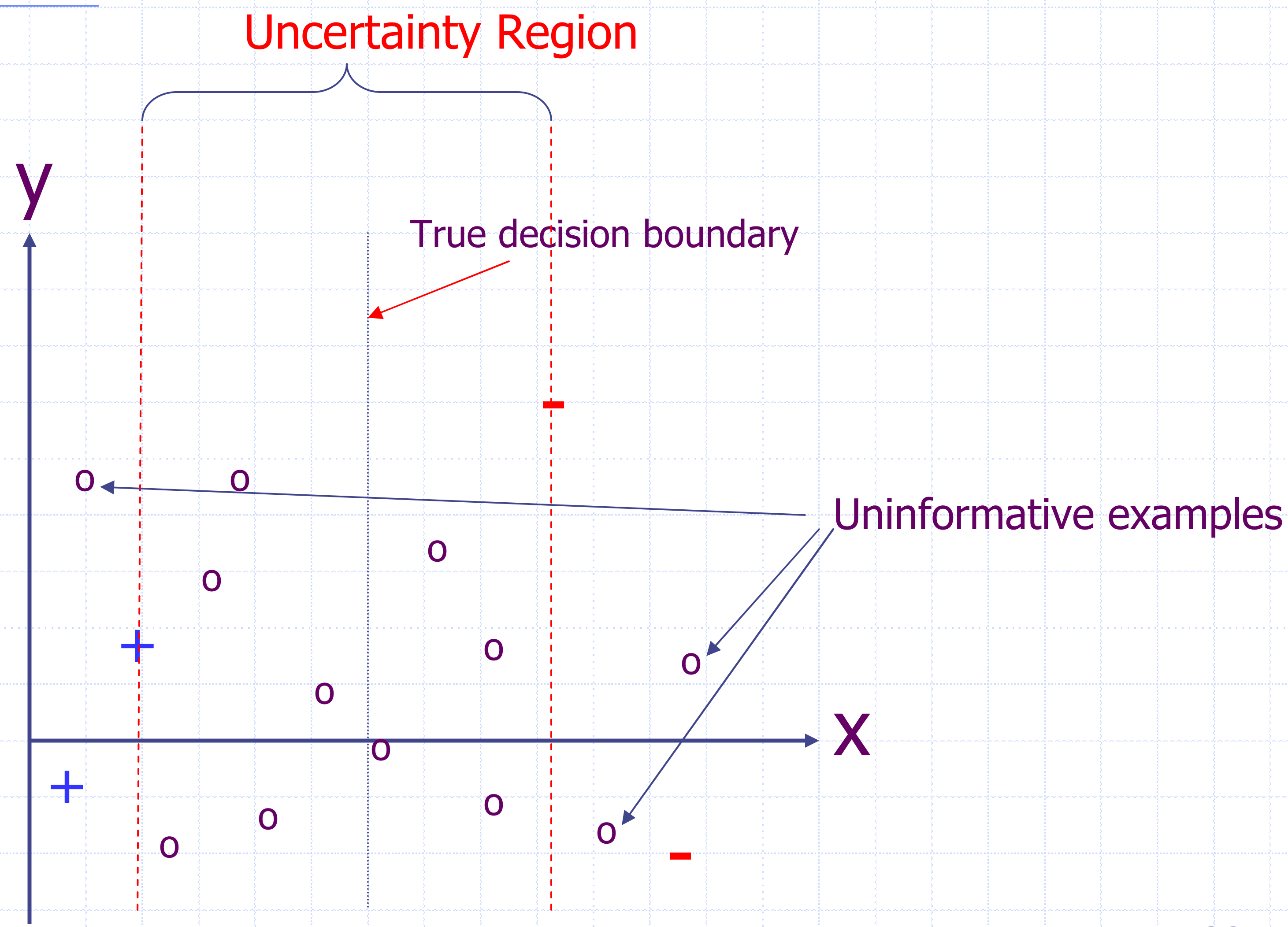
TASK: Locating a boundary on the unit line (x-axis) interval.



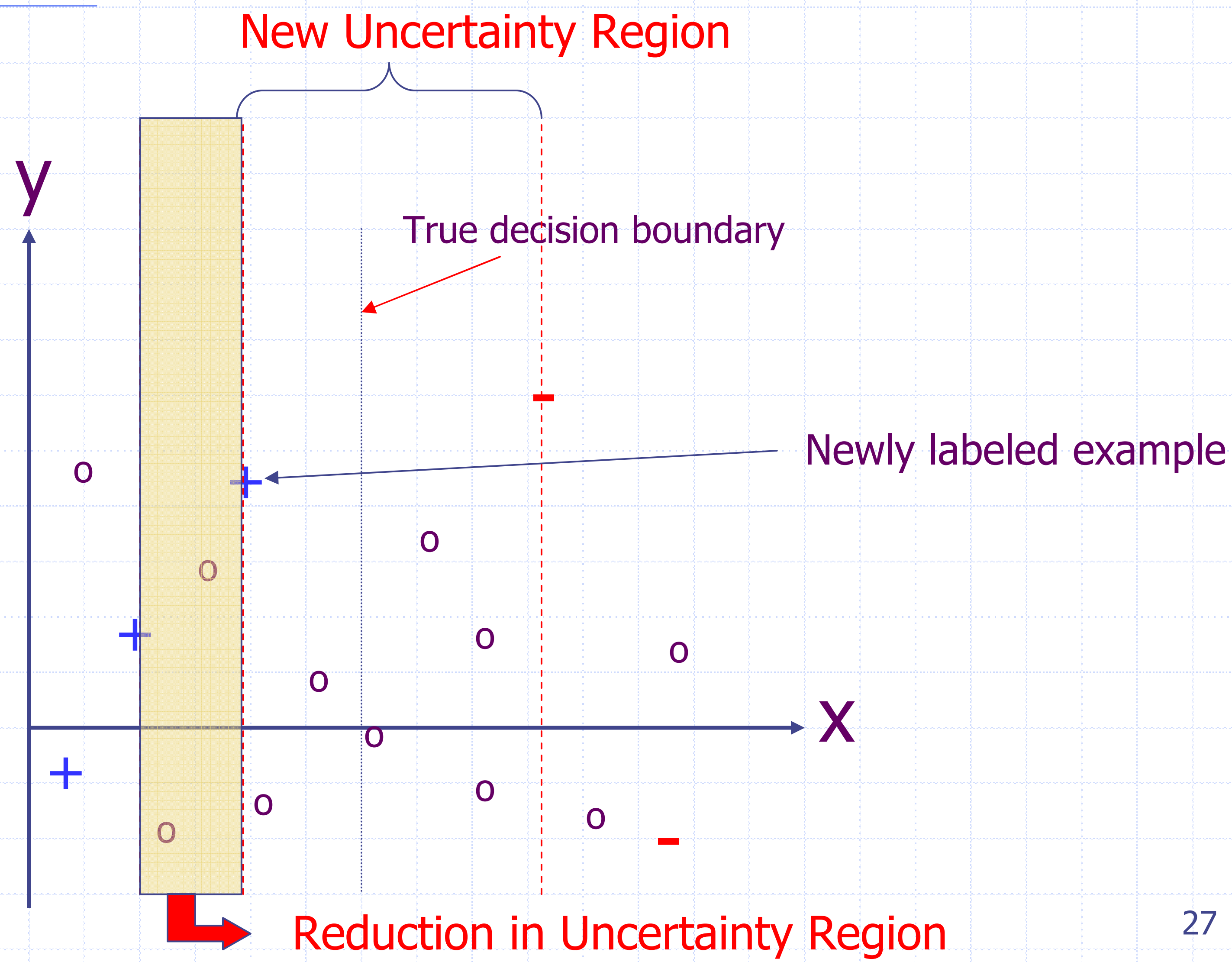
# A Simple Binary Classification Example



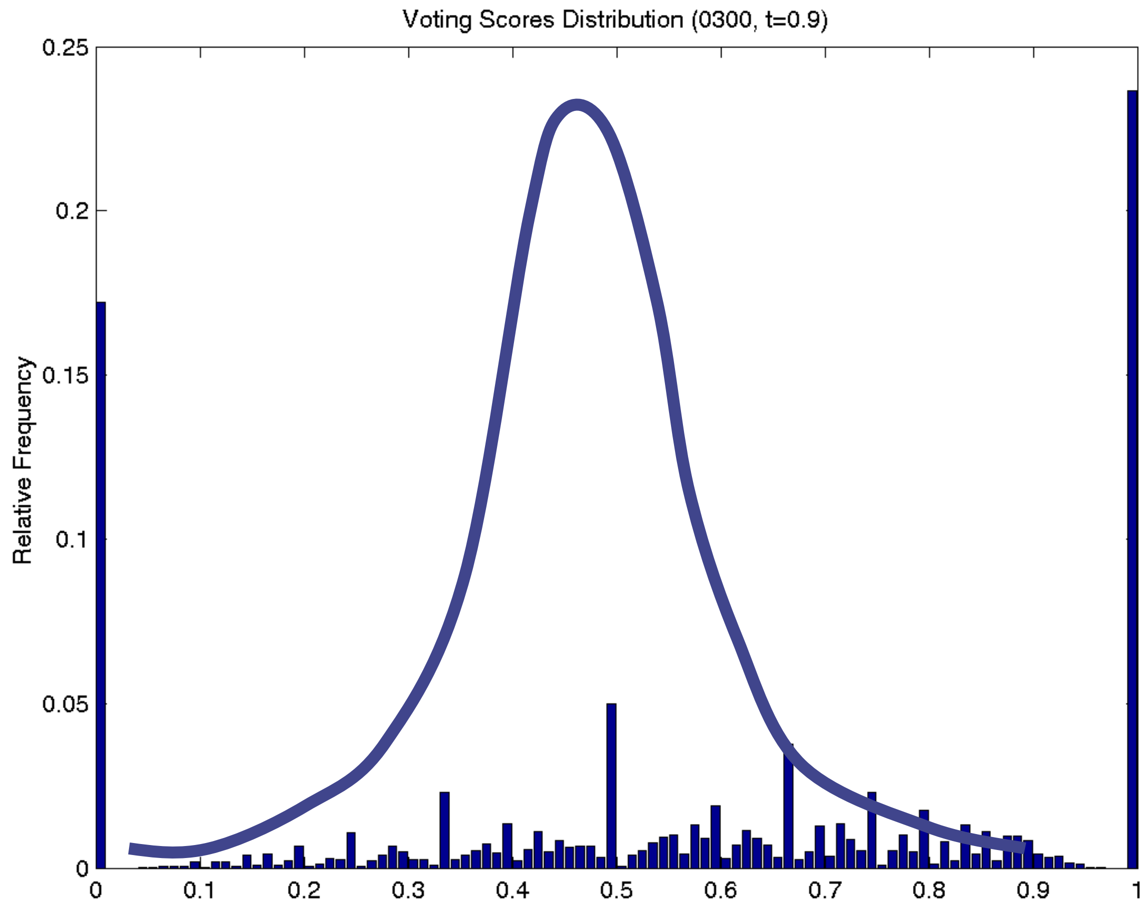
# A Simple Binary Classification Example



# A Simple Binary Classification Example



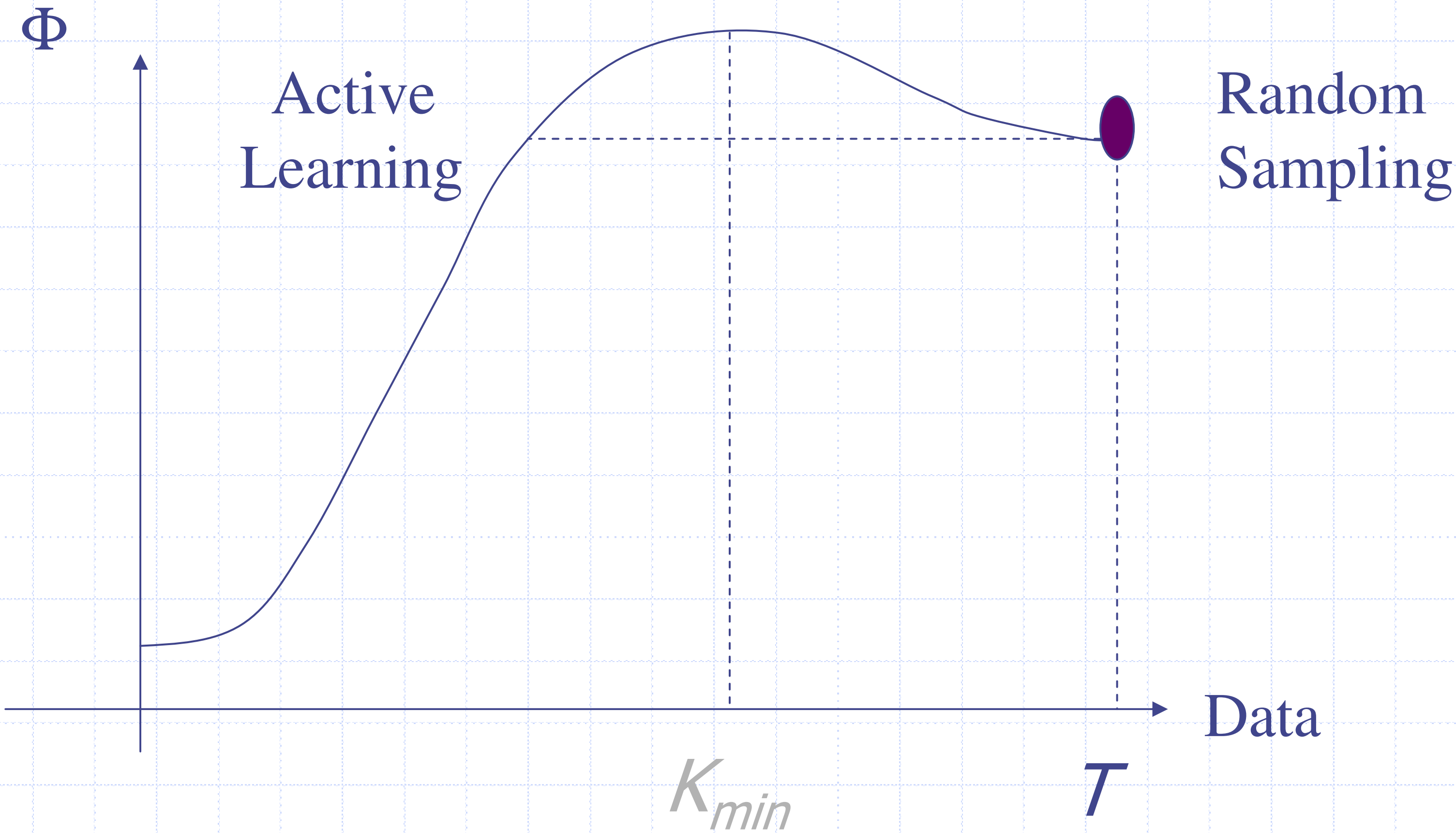
# *Informativeness* of Speech Samples



# Selecting $K_{min}$

*("less is more")*

- Active Learning as optimization problem



# Applications

- Classification Tasks:
  - Text Categorization
  - Call Classification
  - Part of Speech Tagging
  - Word Segmentation
  - Information Extraction
- Automatic Speech Recognition
- Syntactic/Semantic Parsing
- Machine Translation

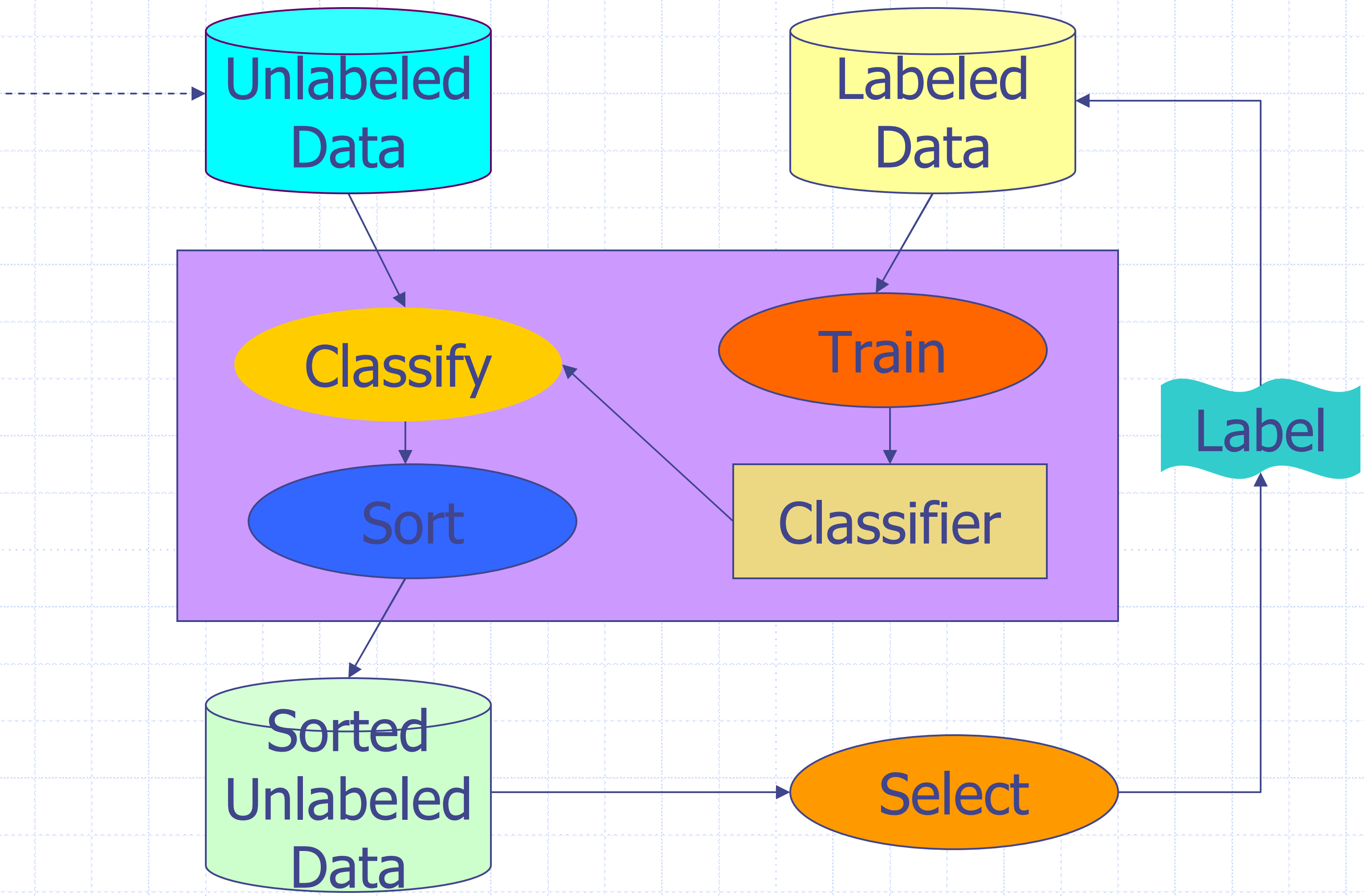
# Outline

- Algorithm Dimension:
  - Passive vs. Adaptive Learning
  - Active Learning
    - Certainty-based
    - Committee-based
  - Unsupervised Learning
  - Combining Active and Unsupervised Learning

# Certainty-based Active Learning for Classification

- Train a base classifier (SVM, Boostexter, etc.)
- While (labelers/data available) do
  - Classify the pool of unlabeled data
  - Sort them according to their informativeness, $I(\Phi)$
  - Select the top $k$ of them
  - Label and add the selected ones to the training data
  - Re-train the classifier
  - Update the pool

# Certainty-Based Active Learning for SLU



# Classification

- *Definition:* The task of assigning objects to 2 or more classes.
- *Example Task / Unit*
  - Part-of-Speech Tagging:
    - Word (e.g. going/VBG)
  - Topic Classification (Text Categorization):
    - Document
  - Call-type Classification:
    - Utterance Transcription (often ASR output)

# Classification Methods

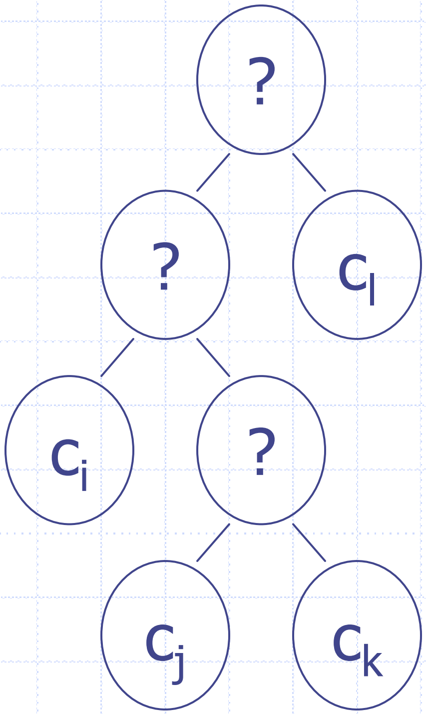
- Rule-based approaches
  - Mostly an expert writing rules for the application based on world/app knowledge
- Machine Learning approaches
  - Employing one of the machine learning algorithms (decision tree, naïve bayes, boosting, SVM, etc.) using the application data
- Hybrid approaches
  - Combining rules with data
  - Learning (probabilities of) rules from data

# Decision Trees

- Classify an object starting from the top node, testing its question, branching to the appropriate node, repeat until it is a leaf.
- Training is based on splitting criterion:
  - Typically *information gain*, which computes the reduction in uncertainty.

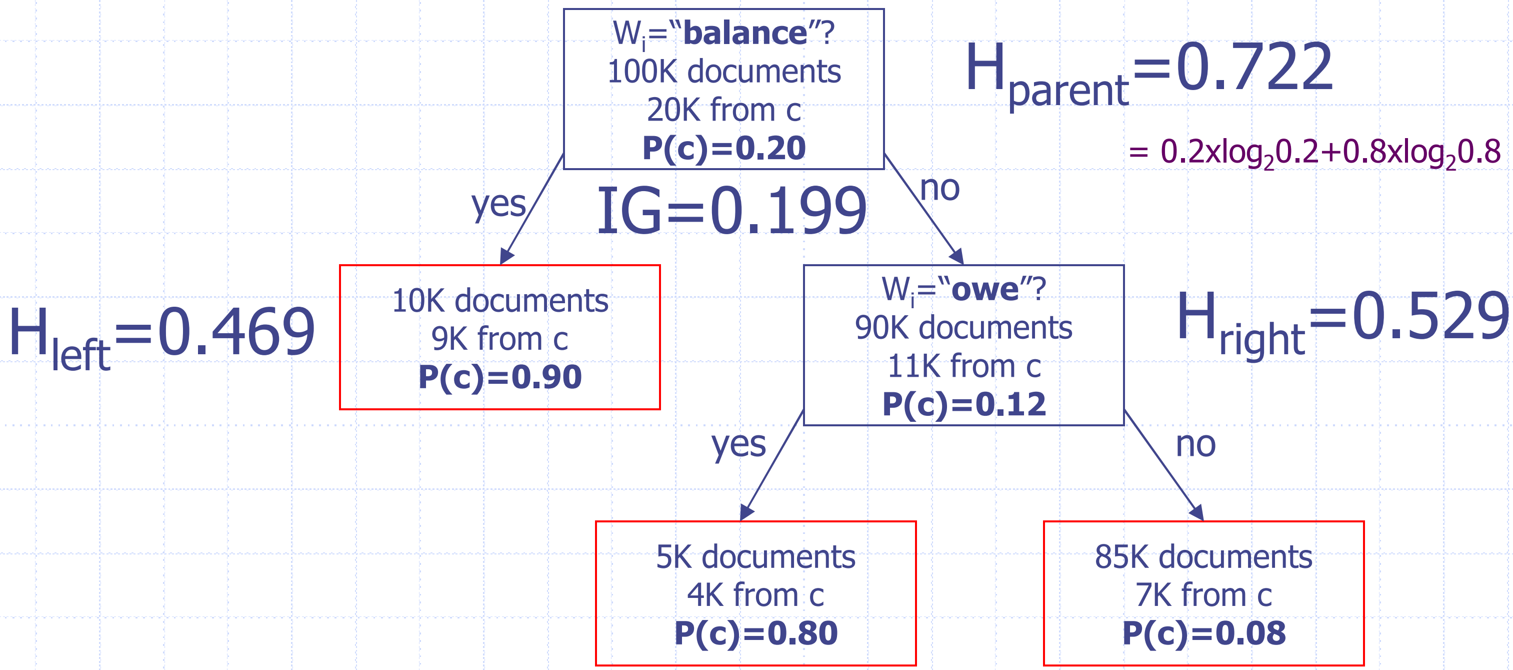
$$G(a) = H(t) - (p_L \times H(t_L) + p_R H(t_R))$$

    where $a$ is the feature, the split is to be decided, $t_{(R|L)}$ is the distribution of the (right|left) node.



# An Example Decision Tree

- Text categorization using a binary classifier with unigram features, deciding whether the class is c (Tellme(Balance)), or not



# Naïve Bayes

- Using the Bayes rule:

$$\hat{c} = \arg\max_{c_i} P(c_i \mid o) = \arg\max_{c_i} \frac{P(o \mid c_i) \times P(c_i)}{P(o)} = \arg\max_{c_i} P(o \mid c_i) \times P(c_i)$$

where $o$ is the object to be classified.

- Assuming conditional independence:

$$P(o \mid c_i) = P(a_1, \ldots, a_n \mid c_j) = \prod_j P(a_j \mid c_i)$$

where $a_j$ is a feature for the object $o$.

# An Example Naïve Bayes Classifier

- Text categorization using unigram features (*bag-of-words*)

$$\arg\max_c P(c \mid sent) = \arg\max_c P(sent \mid c) \times P(c)$$

- Sentence: "balance request"

$$P(sent \mid c) = P(word_1, \ldots, word_n \mid c) = \prod_j P(word_j \mid c)$$

$$score_{c,sent} = P(\text{"}request\text{"} \mid c) \times P(\text{"}balance\text{"} \mid c) \times P(c)$$

$$P(c \mid sent) = \frac{score_{c,sent}}{\sum_i score_{c_i,sent}}$$

# Boosting

- Given the data $(x_1, y_1),\ldots,(x_m, y_m)$ where $x_i \in X, y_i \in Y$
- Initialize the distribution $D_1(i)=1/m$
- For each iteration $t=1,\ldots,T$ do
  - Train a base learner, $h_t$, using distribution $D_t$.
  - Update

$$D_{t+1}(i) = \frac{D_t(i) \times e^{-\alpha_t \times y_i \times h_t(x_i)}}{Z_t}$$

where $Z_t$ is a normalization factor and $\alpha_t$ is the weight of the base learner, computed using the error rate of that learner.

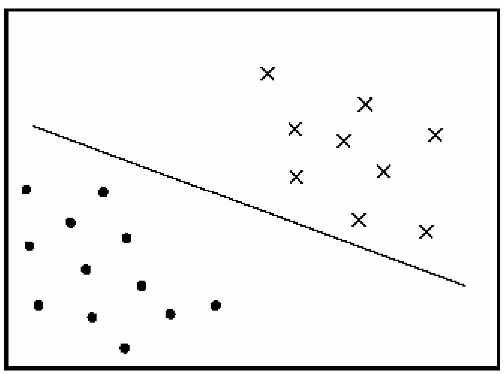
- The output of the final classifier is defined as:

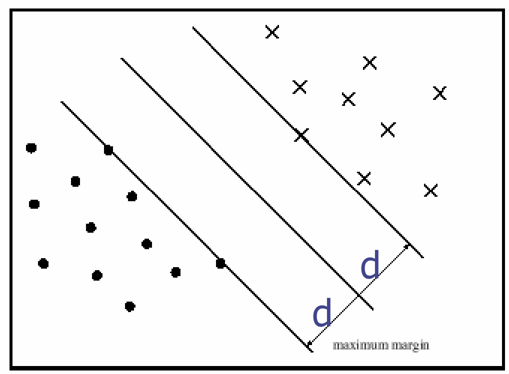
$$f(x) = \sum_{t=1}^{T} \alpha_t \times h_t(x)$$

$$H(x) = sign(f(x))$$

# Support Vector Machines

- Given a set of examples belonging to two different classes, the Support Vector Machine (SVM) tries to separate them with the maximum margin (Vapnik).





# Evaluation Metrics

$$\text{Accuracy} = \frac{\#\text{correctly\_classified}}{\#\text{examples}}$$

Classification Error Rate (CER) = 1 - Accuracy

- Assuming thresholding using the scores

| | decision is correct | decision is incorrect |
|---|---|---|
| Score>=Threshold (accept) | a | b |
| Score<Threshold (reject) | c | d |

$$\text{Recall} = \frac{a}{a+c} = \frac{\#\text{correct and accepted}}{\#\text{correct}}$$

$$\text{Precision} = \frac{a}{a+b} = \frac{\#\text{correct and accepted}}{\#\text{accepted}}$$

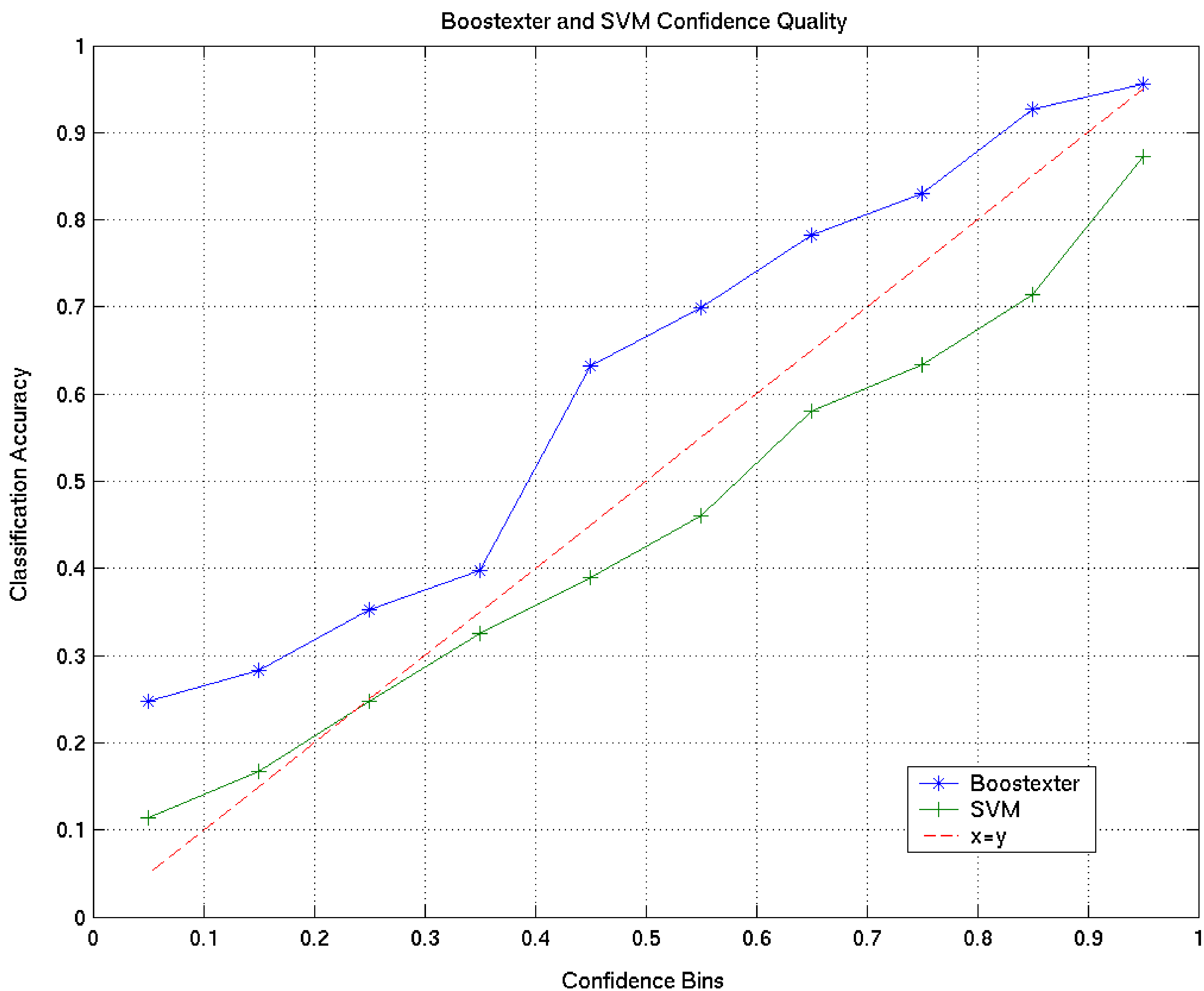
$$\text{F-Measure} = \frac{\text{Recall} \times \text{Precision}}{\alpha \times \text{Recall} + (1-\alpha) \times \text{Precision}}$$

$$\text{False-Rejection} = \frac{c}{c+d} = \frac{\#\text{correct and rejected}}{\#\text{rejected}}$$

$$\text{False-Acceptance} = \frac{b}{a+b} = \frac{\#\text{wrong and accepted}}{\#\text{accepted}}$$

# Error Modeling

- Needs an informativeness measure to sort the candidate unlabeled utterances
- Use confidence scores output by the learners.
- e.g. for the Naïve Bayes classifier, it is nothing but $P(c_i | o)$
- Alternative usages:
  - Confidence of the top scoring class (e.g. $\max_i P(c_i | o)$)
  - Difference in the confidences of top two scoring classes
  - KL(P($C$|$X$)||P($C$))


Boostexter and SVM Confidence Quality
Classification Accuracy
Confidence Bins
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Boostexter
SVM
x=y

# Selected Bibliography for Certainty-Based Active Learning


- Lewis and Catlett, ICML'94 (*Text Categorization*)
- Cohn et al., ML'94 (*Text Categorization*)
- Thompson et al., ICML'99 (*Parsing and Info. Ext.)*
- Schohn and Cohn, ICML'00 (*Text Categorization*)
- Hwa, EMNLP/VLC'00 (Parsing)
- Hakkani-Tür et al., ICASSP'02 (*ASR*)
- Tang et al., ACL'02 (*Parsing*)
- Sassano, ACL'02 (Japanese Word Segmentation)
- Tur et al., ICASSP'03 (*Call Classification*)

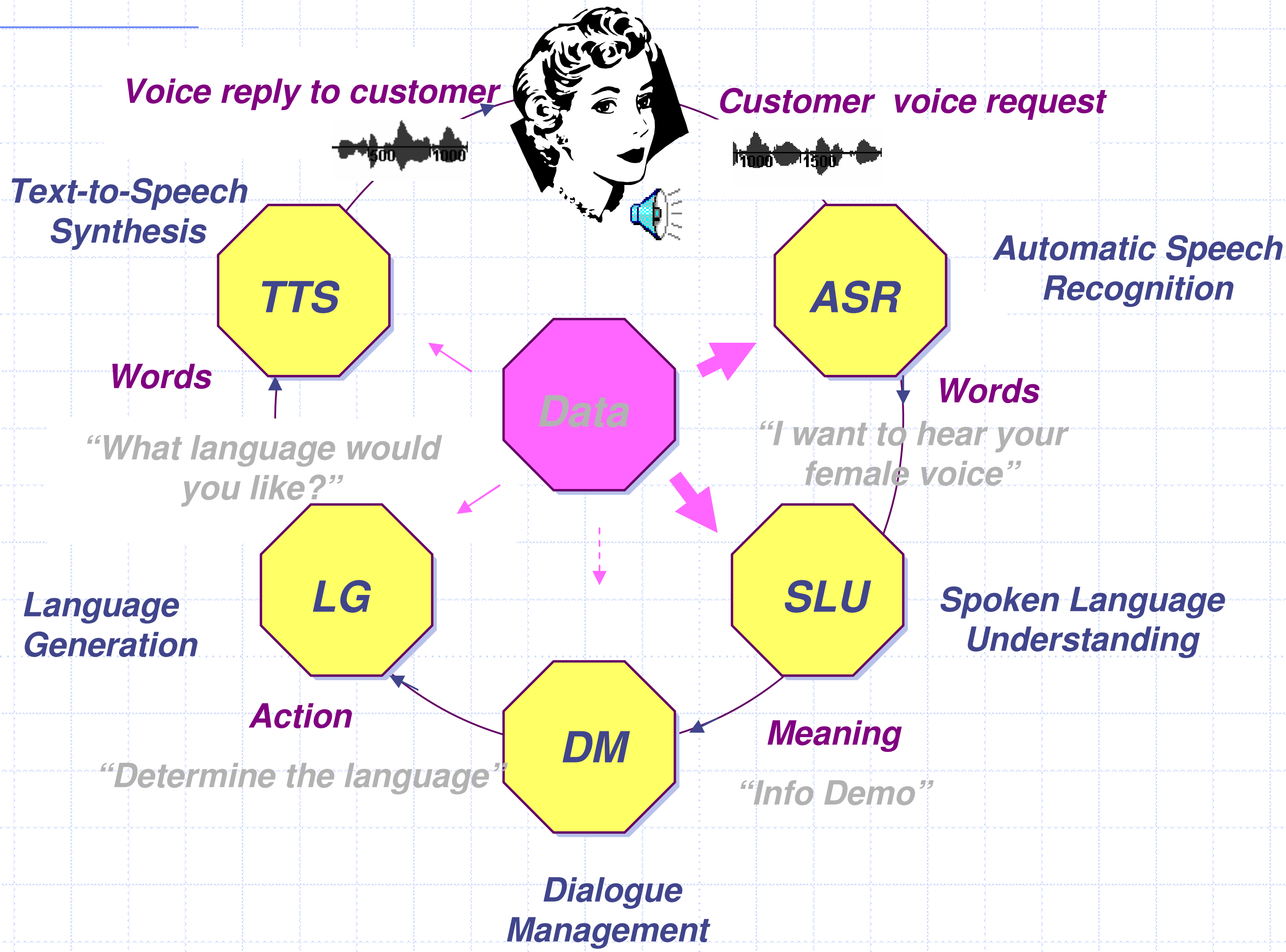
# Text Categorization

- *Lewis and Catlett ICML'94*
- AP articles, 10 classes
- Classifier: Decision Trees
- Used a simple probabilistic classifier for sample selection
- Reduced the amount of human-labeled data needed by a factor of 10.

# Parsing

- (Hwa, EMNLP/VLC, 2000)
- Criterion: Tree Entropy (TE)
  - Parse the sentence, *s*
    - i.e. get multiple parse trees, $v \in V$, with confidences, $p(v)$
  - Compute $TE(s) = -\sum_{v \in V} p(v) \log p(v)$
  - Pick the sentences with high TE values
- Decreased the amount of training data needed to achieve the same performance by 36%

# Human-Machine Spoken Dialog



# Conversational Speech

- How May I Help You?
- hello [ uh ] [ .clrt ] excuse me I I would like I don't understand my **bill** I
- Okay. What is your question?
- what is my what
- I'm sorry, I didn't **understand** that. How may I help you?
- well [ eh ] I don't understand certain **items** on my **bill** like [uh] [.lps] it says **summary toll calls** [.clrt] excuse me 87 cents now I get listed for **toll calls** th- [ eh ] there's [ uh ] [ um ] [ .lps ] there's a whole list of [uh ] **toll calls** that I made why do they put this one separately…

# Voice-Enabled Services Complexity

**Simple ASR; isolated words, connected digits**

**Command And Control**

(e.g., Simple call Routing; VRCP; Voice dialing)

AT&T VRCP

**Larger vocabulary, defined grammars**

**Prompt Constrained Natural Language**

(e.g., Travel Reservations, Finance, Directory asst)

E*Trade

United Airlines

**Very large vocabulary, NL, DM, TTS**

**Free-form Natural Language Dialogue**

(Customer Care, Help Desks, E-Commerce)

ACS 0300

IRS

***Complexity and Functionality***

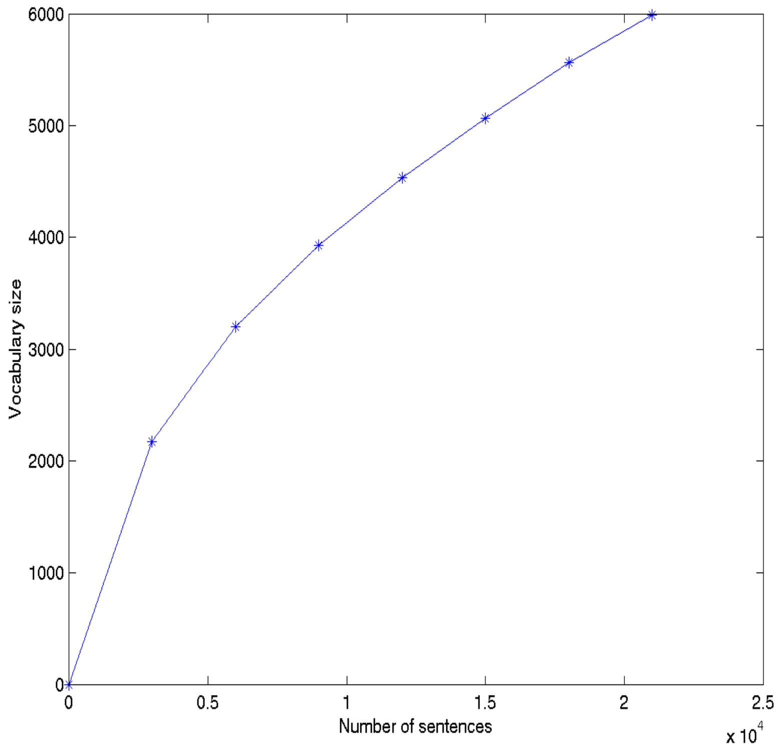
1990 → 2002

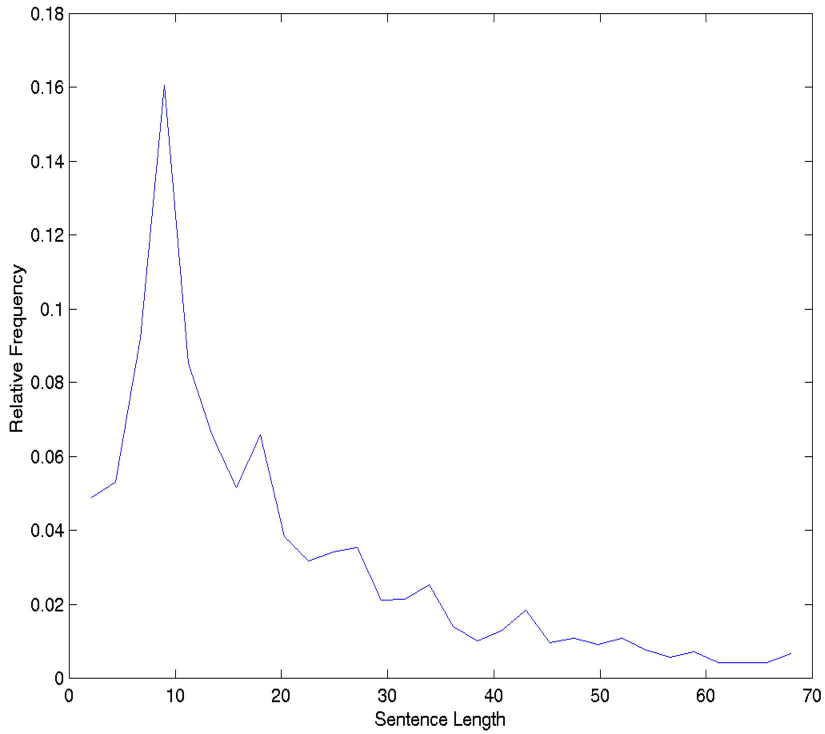
# Data Driven Learning

(Speech and Language)

- **Input:** Speech Utterance $u_i$
- Automatic Speech Recognition
  - Gaussian Mixture Modeling (HMMs)
  - N-gram estimations ($P(w_i|w_{i-n+1}, ..w_{i-1})$)
- Semantic Associations
  - $T=\{w_i,c_j\}$
  - Feature Extraction ($\#(f_k,c_j)$)
    - (Salient) N-grams $\rightarrow$ Bayes,Boosting, SVM Classifiers)
- **Output:** Model $\lambda$
  - Speech recognition: $\lambda_{ASR}: u \rightarrow w$
  - Semantic Associations: $\lambda_{NL}: w \rightarrow c$

# Corpus Statistics



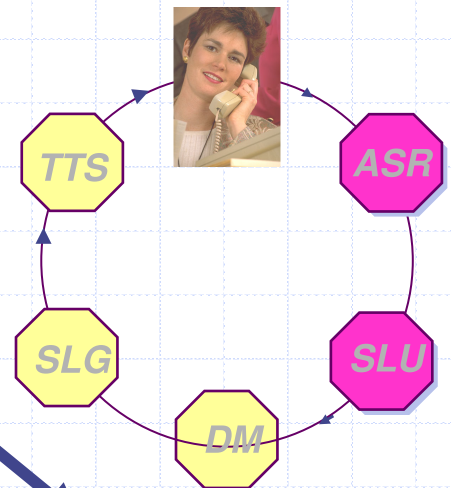



# Ways to say “question about my bill”

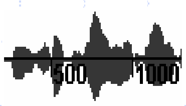
105 question about my bill
63 question on my bill
57 calling about my bill
43 talk to somebody about my bill
41 talk to someone about my bill
32 questions about my bill
30 problem with my bill
23 speak to someone about my bill
22 calling about a bill
20 calling about my phone bill
16 questions on my bill
16 question about a bill
15 talk about my bill
11 question about my phone bill
11 question about my billing
11 discuss my bill
10 speak with someone about my bill
10 calling about my billing
9 problem with my phone bill
9 calling about my telephone bill
8 speak to someone in billing
8 question about the bill
7 speak to somebody about my bill
7 speak to a billing
7 question on my phone bill
7 calling regarding my bill
7 calling concerning my bill
6 talk to somebody in billing
6 questions about my billing
6 question on my billing
6 problem with my billing
6 information about my bill
6 calling about my A T and T bill
5 talk to someone about my phone bill
5 talk to someone about a bill
5 talk to somebody about my billing
5 talk to somebody about a bill
5 speak to someone in the billing
5 speak to someone about a bill
5 questions on my billing
5 question on the bill
5 question on a bill
5 question my bill
5 calling in regards to my bill
5 calling about the bill
4 talk to someone about my telephone bill
4 talk to somebody about my account
4 talk to billing
4 speak with someone in billing
4 question about my telephone bill
4 information on my bill
4 calling regarding my statement
..............
1 talk to someo- to someone about my moms telephone bill
1 question about the new A T and T billing
1 calling for Bertha Fitz******* about a b- statement

Total 1083 variations in 1912 matches

# Speech Recognition and Understanding Labeling Process




S(t) → **Utterance Transcription** → T → **Semantic Annotation** → A




"Tell me my balance for the business and residential account"

Request(Balance(Business)), Request(Balance(Residential))

# Basic Formulation of ASR

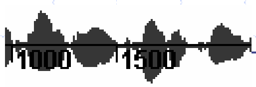
- Given an acoustic observation sequence $\mathbf{X}=X_1,X_2,\ldots,X_n$ and a specified word sequence $\hat{\mathbf{W}} = w_1 w_2 \ldots w_m$ , then

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{W} \mid \mathbf{X}) = \arg\max_{\mathbf{W}} \frac{P(\mathbf{W})P(\mathbf{X} \mid \mathbf{W})}{P(\mathbf{X})} = \arg\max_{\mathbf{W}} P(\mathbf{W})P(\mathbf{X} \mid \mathbf{W})$$

- $P(\mathbf{X}|\mathbf{W})$ is the acoustic model
- $P(\mathbf{W})$ is the language model

# ASR - Overview

Given the acoustic observation sequence $A=a_1,a_2,\ldots,a_m$, what is the most probable word sequence $W=w_1,w_2,\ldots,w_n$?



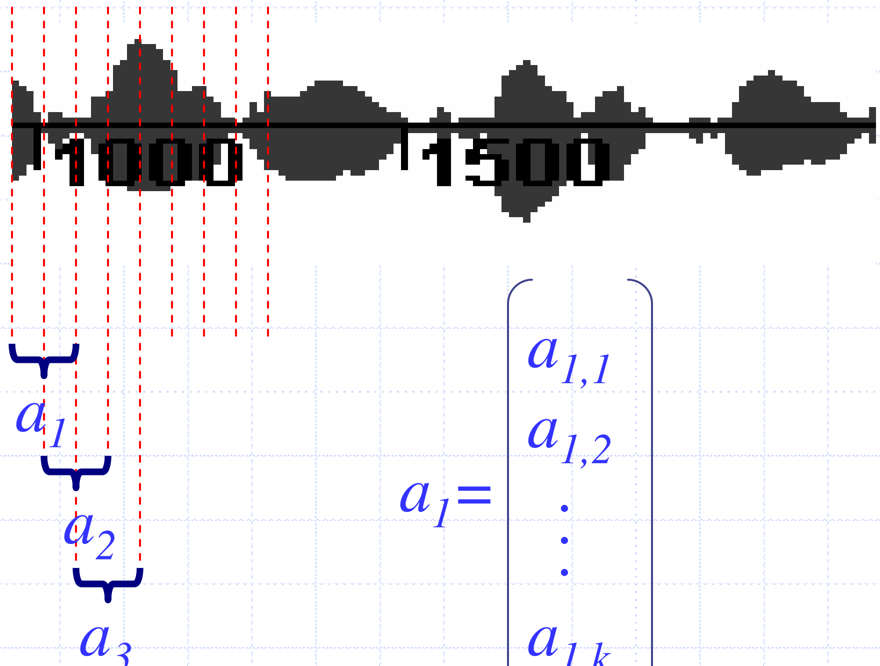

→ Feature Extraction → $A$ → Pattern Matching → $\hat{W}$

$$\hat{W} = \arg\max_{W} P(W \mid A) = \arg\max_{W} \frac{P(A \mid W)\,P(W)}{P(A)}$$

$$= \arg\max_{W} \underbrace{P(A \mid W)}_{\text{Acoustic Model}}\,\underbrace{P(W)}_{\text{Language Model}}$$

# Feature Extraction

- Extract features from the speech signal that are relevant for recognition.



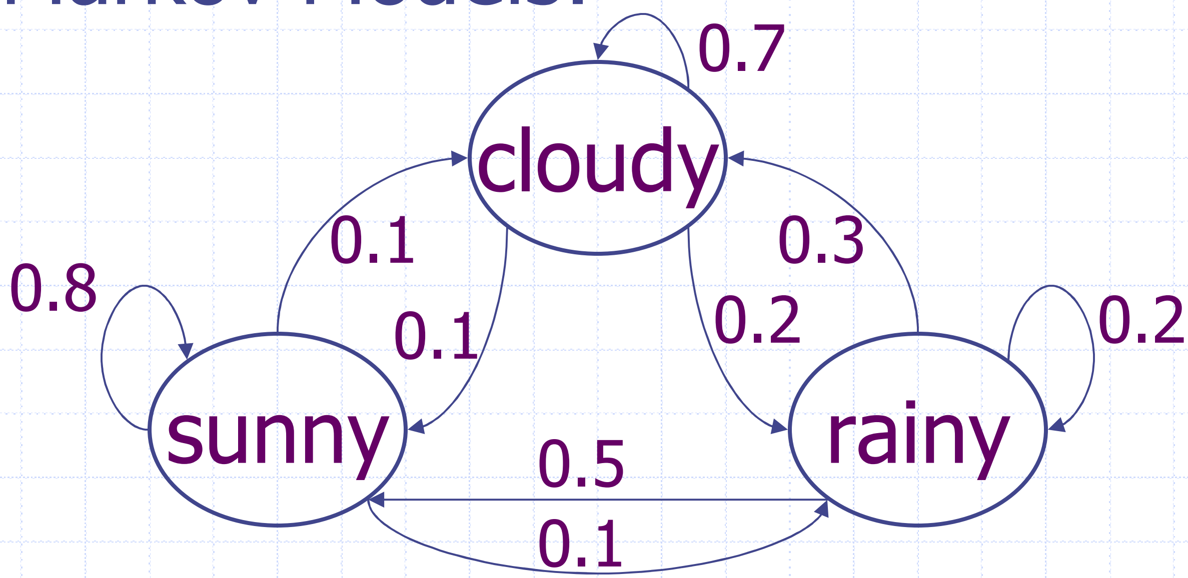

$$a_1 = \begin{pmatrix} a_{1,1} \\ a_{1,2} \\ \vdots \\ a_{1,k} \end{pmatrix}$$

# Acoustic Modeling

- *P(A/W)*
- To extract sub-word units from the acoustic features.
- State-of-the-art systems are based on the use of Hidden Markov Models (HMMs).
- For an extensive discussion of HMMs, see Rabiner 1989.

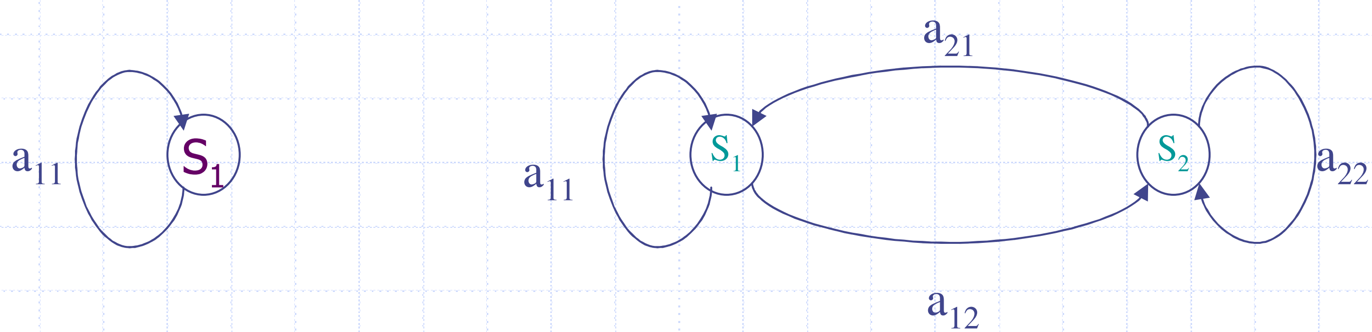
# A Very Brief Introduction to HMMs

- Markov Models:




- $\Pi(\text{cloudy})=0.2$
- O=cloudy cloudy rainy sunny
- $P(O|\text{model})=0.2\times0.7\times0.2\times0.5=0.014$

# Hidden Markov Models




- Observations are probabilistic functions of the states.
- Additional Elements:
  - $B=\{b_i(o_j)\}$, the observation symbol probabilities, for observing $o_j$ at state i.
  - e.g.: $b_1(\text{sunny}) = 0.3$

# Observation Evaluation

- What is the probability of the observation sequence, O, given the model parameters?

1. Initialization:

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \le \mathrm{i} \le \mathrm{N}$$

2. Induction:

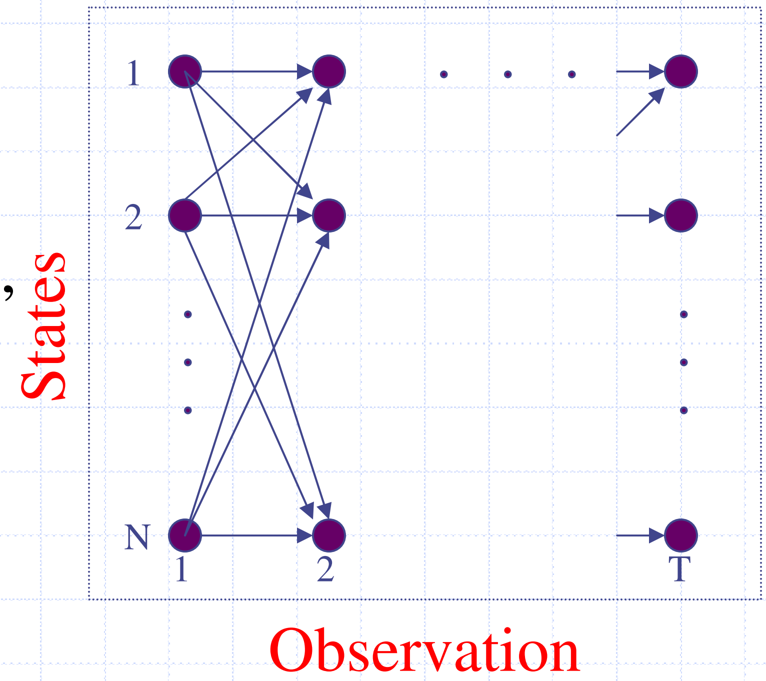
$$\alpha_{t+1}(j) = (\sum_{i=1}^{N} \alpha_t(i) a_{ij}) b_j(o_{t+1}),$$

$$1 \le \mathrm{t} \le \mathrm{T}-1,\ 1 \le \mathrm{j} \le \mathrm{N}$$

3. Termination:

$$P(O \mid \Phi) = \sum_{i=1}^{N} \alpha_T(i)$$

Trellis



# Other HMM Problems

- The Viterbi Algorithm: What is the most probable state sequence, given the observation sequence, O, and model parameters $\Phi=(A,B,\Pi)$?
- The Baum-Welch Algorithm: How do we adjust the model parameters $\Phi=(A,B,\Pi)$, to maximize $P(O/\Phi)$, $O=o_1,\ldots,o_T$?

# Language Modeling

- Probability of word sequences.
- $W$= "I wanna fly to Boston"

$$P(W) = P(\text{I}) \times P(\text{wanna} \mid \text{I}) \times \ldots \times P(\text{Boston} \mid \text{I, wanna, fly, to})$$
$$= P(\text{I}) \times P(\text{wanna} \mid \text{I}) \times \ldots \times P(\text{Boston} \mid \text{to})$$

- Maximum likelihood estimates

$$P(\text{Boston}) = \frac{C(\text{Boston})}{N} \qquad P(\text{Boston} \mid \text{to}) = \frac{C(\text{to, Boston})}{C(\text{Boston})}$$

- $C(w_i, \ldots, w_j)$ is the number of times word sequence $w_i, \ldots, w_j$ occurs in the training text.

# Smoothing

- What about the word sequence:

  $W=$"I wanna fly to Geneva"

  if $C$(to,Geneva) = 0, as it never occurred in the training set?
- Aim: To assign a non-zero probability to previously unseen sequences.
- Robustness to unseen data.

# Smoothing - Approaches

- Add One

$$P_{smooth}(w_i) = \frac{C(w_i)+1}{N+V} \qquad P_{smooth}(w_i \mid w_{i-1}) = \frac{C(w_{i-1}, w_i)+1}{C(w_{i-1})+V}$$

- Interpolation

$$P_{smooth}(w_i \mid w_{i-1}) = \lambda \times P(w_i \mid w_{i-1}) + (1-\lambda)P(w_i)$$

- Back-off

$$P_{smooth}(w_i \mid w_{i-1}) = \begin{cases} P(w_i \mid w_{i-1}), & \text{if } C(w_{i-1}, w_i) > 0 \\ \alpha \times P(w_i), & \text{otherwise} \end{cases}$$

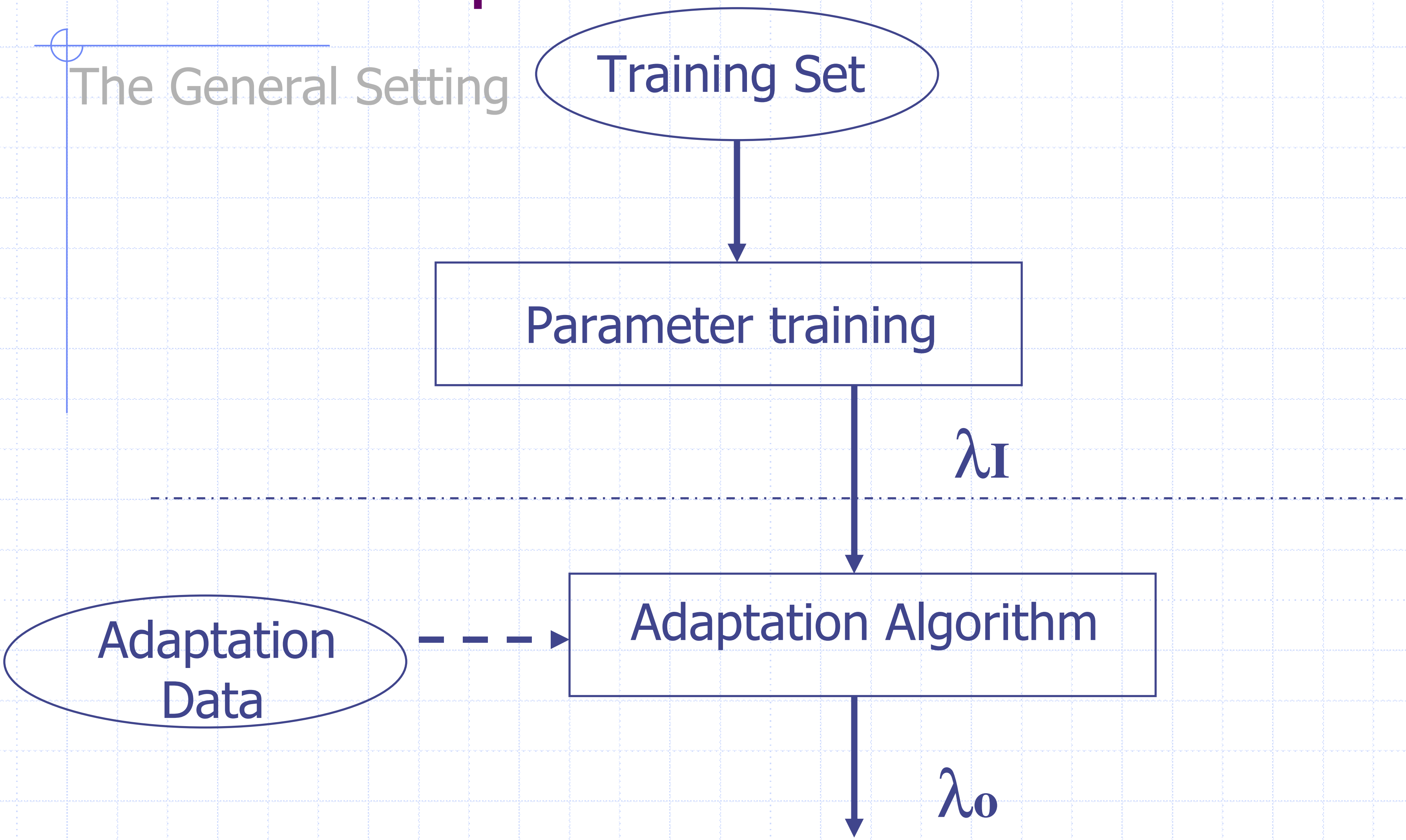
# Adaptation

- Robustness to mismatched conditions, like variations in the:
  - Microphone
  - Environment noise
  - Speaker
  - Topic, etc.

  e.g.: Speaker dependent versus speaker independent systems.

# Model Adaptation

## The General Setting



# Adaptation Schemes

## Example: Language Modeling

- Interpolated Model

$$P(w_i \mid h) = \alpha(h)P_I(w_i \mid h) + (1 - \alpha(h))P_A(w_i \mid h)$$

- Cache Language Models

$$P_{cache}(w_i \mid w_{i-n+1} \ldots w_{i-1}) = \lambda_c P_s(w_i \mid w_{i-n+1} \ldots w_{i-1}) + (1-\lambda_c)P_{cache}(w_i \mid w_{i-2}w_{i-1})$$

# Acoustic Model Adaptation

- Maximum a Posteriori (MAP)
  - Consider also the prior distribution for the parameters of the model.

$$\hat{\Phi} = \arg\max_{\Phi} P(\Phi \mid W) = \arg\max_{\Phi} P(W \mid \Phi) P(\Phi)$$

  - Useful when the adaptation data is limited.
- Maximum Likelihood Linear Regression (MLLR)
  - A linear transformation of the model parameters are estimated.

# Language Model Adaptation

- Cache-based Language Models

$$P(w_i \mid w_{i-1}) = \lambda \times P_{cache}(w_i \mid w_{i-1}) + (1-\lambda) \times P_{global}(w_i \mid w_{i-1})$$

  - $P_{cache}(w_i/w_{i-1})$ is estimated from a cache, which contains the most recently dictated words.
- Topic Adaptation
  - Build topic dependent language models from the topic clusters.
  - Interpolate the topic dependent models.
- Dialog state dependent language models
  - Build a state dependent model using the previous responses to the current" prompt.

# ASR - Evaluation

- Word Error Rate (WER)

$$WER = \frac{\# \text{Ins} + \# \text{Del} + \# \text{Subs}}{\# \text{Ref. Words}}$$

REF: i'd like to review my services that i have
HYP: i'd like to have a review the services i have

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| REF: | i'd | like | to | **** | * | review | MY | services | THAT | i | have |
| HYP: | i'd | like | to | HAVE | A | review | THE | services | **** | i | have |

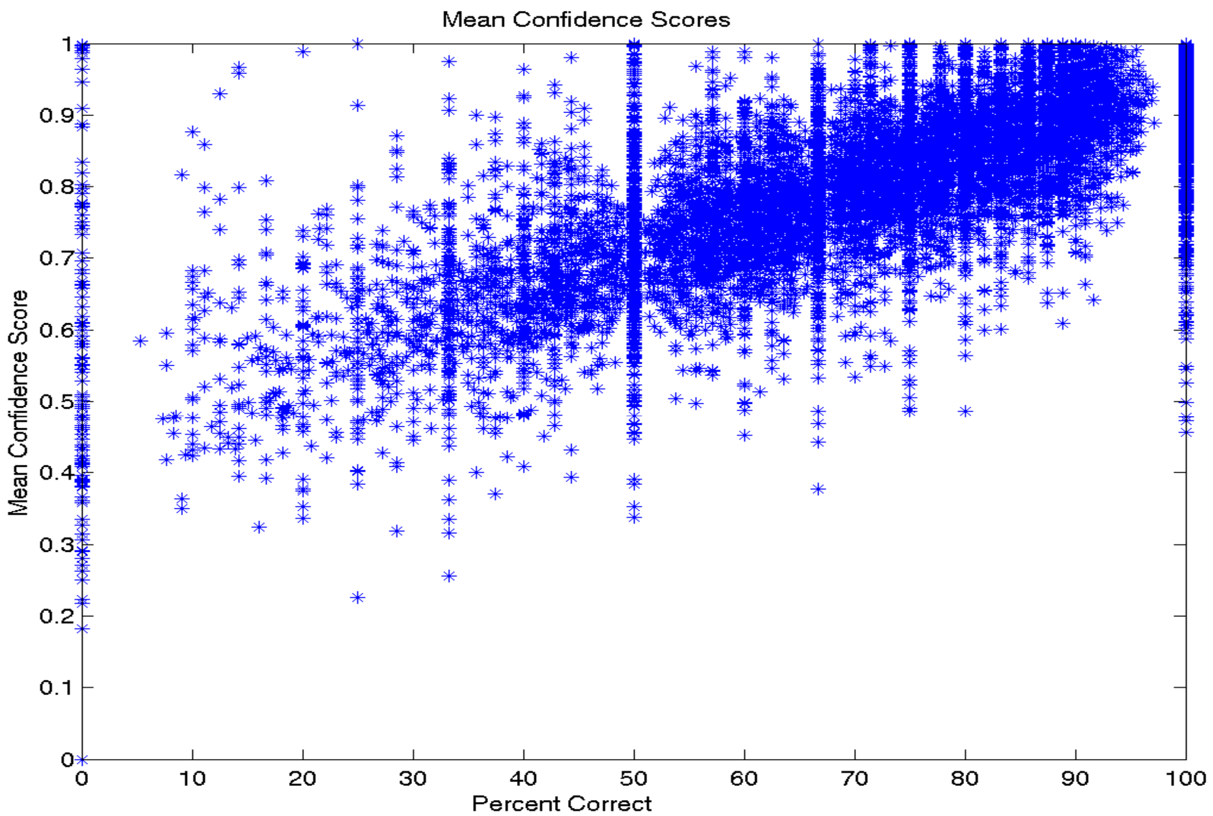
Insertions Substitution Deletion

- Word Accuracy (WA)

$$WA = 1 - WER$$

# ASR Confidence Scores

- Probability of utterance $u_i$ being correctly recognized by current model $\lambda$



# ASR Confidence Scores

- Mark each phone/word/utterance with a score of confidence.
- ASR word confidence scores for
  - Selective Sampling for Active Learning
  - Probability Estimation for Unsupervised Learning
  - Selective Sampling for Unsupervised Learning
- Word confidence scores and word confusion networks (sausages) for improving
  - natural language understanding
  - machine translation
  - named entity extraction

# Likelihood Ratio Tests

- Likelihood ratio (LR) test (Lleida and Rose, 1996)

$$LR(A, \lambda^c, \lambda^a) = \frac{P(A \mid \lambda^c)}{P(A \mid \lambda^a)} \underset{H_1}{\overset{H_0}{\gtrless}} \tau$$

  - A: a sequence of feature vectors
  - $\lambda^c$: target model
  - $\lambda^a$: alternative model
- Word level confidence scores are obtained by combining LR scores.
- Requires training.

# Word Graph Based Approaches

- Word-Graph-based Approaches
  - Derived from the lattice output of ASR.
  - No need for training
- ASR lattices → Sausages (word confusion networks)
  - (Mangu, *et al.*, 2000)
  - Word posterior probability estimates on the sausages → word confidence scores
- (Hakkani-Tür and Riccardi, 2003)

# Hybrid Approaches

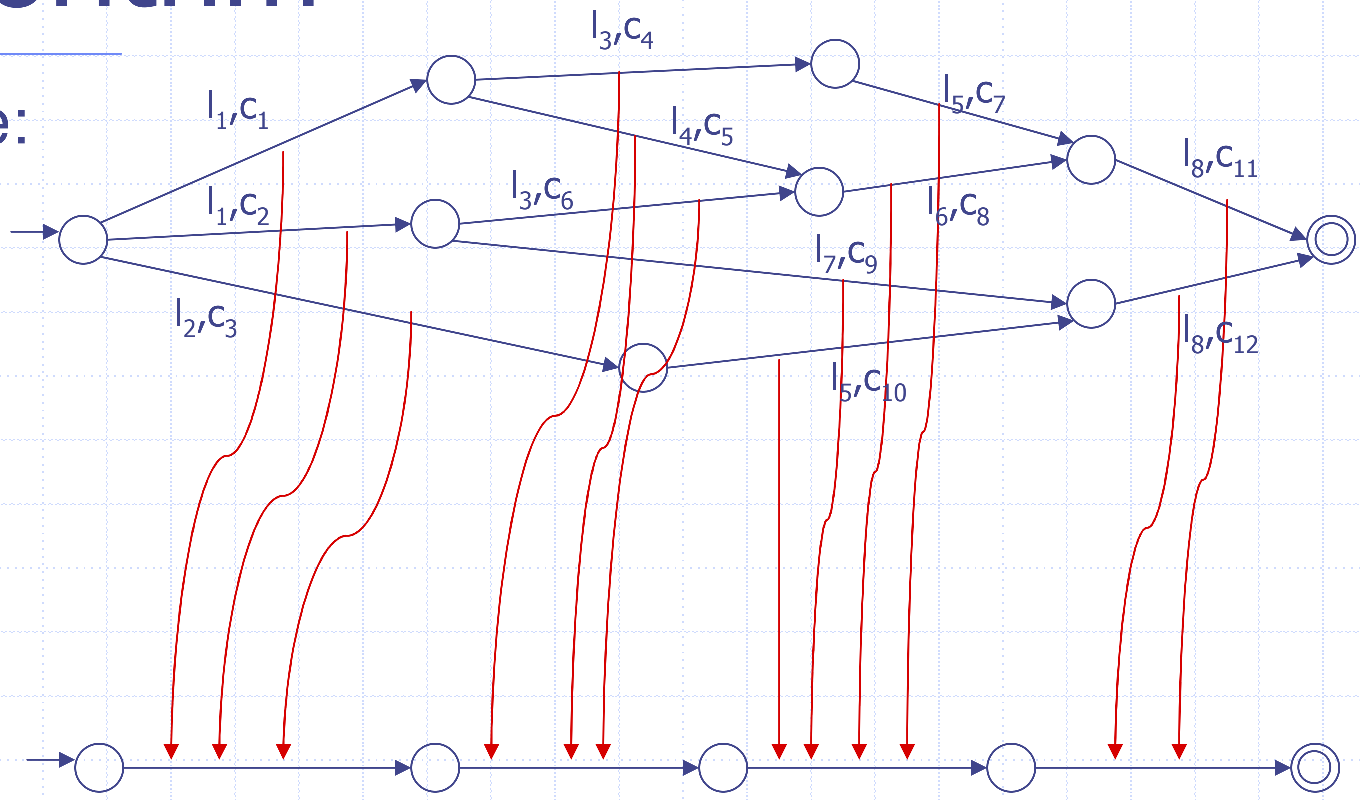
- Approaches that use:
  - Acoustic features
  - Word lattice features
  - Linguistically motivated features

  to come up with word confidence scores (*eg*: Zhang and Rudnicky, 2001)
- Requires training.

# Algorithm

Lattice:

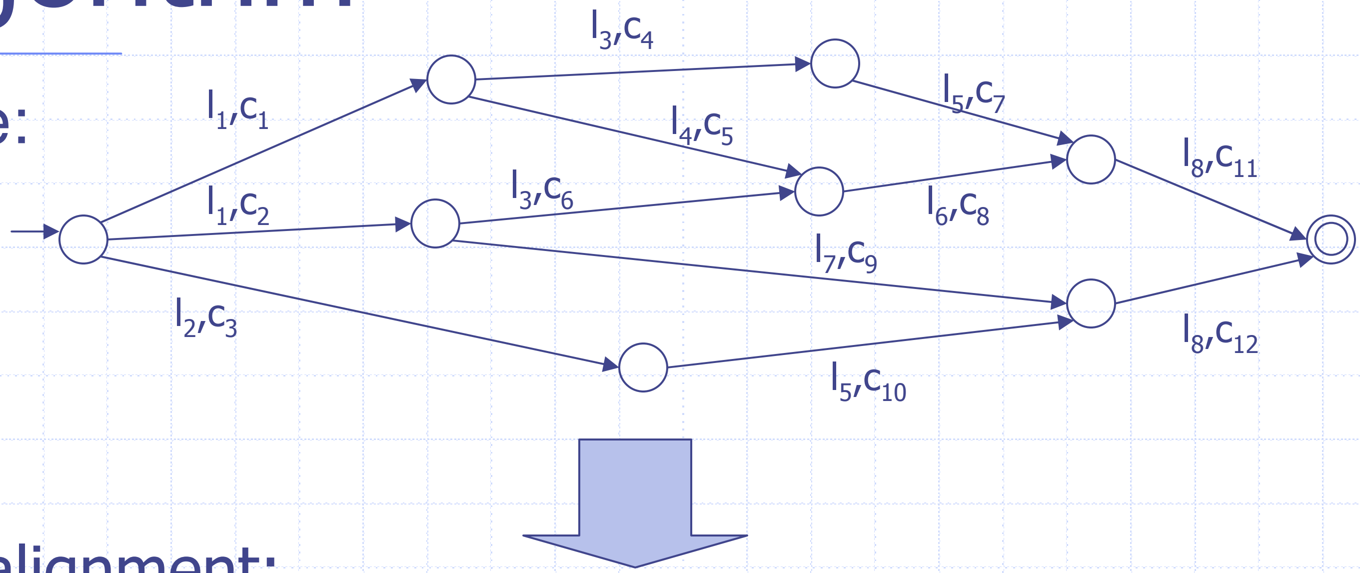
Pivot:



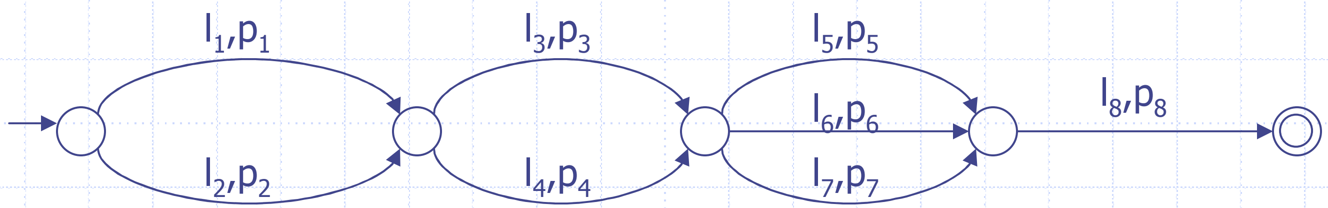

$l_i$ : labels
$c_i$: costs
$p_i$: posterior probabilities

# Algorithm

Lattice:




Pivot alignment:




$l_i$ : labels
$c_i$: costs
$p_i$: posterior probabilities

# Algorithm

- **Compute** the posterior probabilities of all transitions on the lattice
- **Select** a path as a baseline
  - [random/best/longest path]
- **For** all transitions in the lattice,
  - Find the most overlapping position (wrt start and ending state times) on the pivot/baseline
  - If a transition with same label already occurs there, increment its posterior
  - Otherwise, insert a new transition to the pivot/baseline

# Algorithm Details

- Time information is not necessary, but beneficial.
  - Time info is estimated as approximate state location.
- The labels on arcs can be words, phones, semantic tags, etc.
  - E.g. slot confidence scores
- Algorithmic complexity:O($N*M$)
  - MEMORY: smaller than word lattices (7% of lattices).
  - TIME: much faster than sausage computation of Mangu et al. (2000), which runs in $O(N^3)$.

$N$: Number of arcs in the lattice

$M$: Number of arcs on the best/longest/random path.

# Evaluation of Confidence Scores

- Test Set: 2,174 utterances (~31K words) form AT&T HMIHY?SM spoken dialog system test data.
- Baseline: Best Path
- Select a threshold, accept as correct recognition if confidence score is bigger than threshold.
- False Acceptance Rate (*FA*)

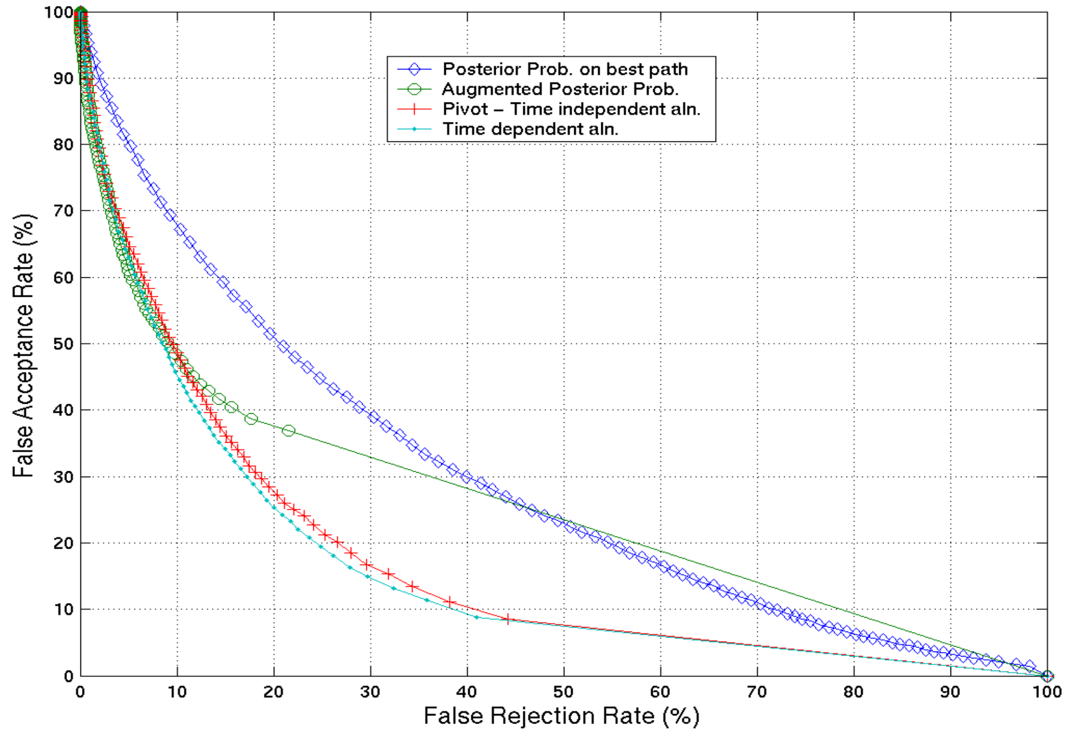
$$FA = \frac{\text{\# of misrecognized words that are accepted}}{\text{\# of words that are accepted}} \times 100\%$$

- False Rejection Rate (*FR*)

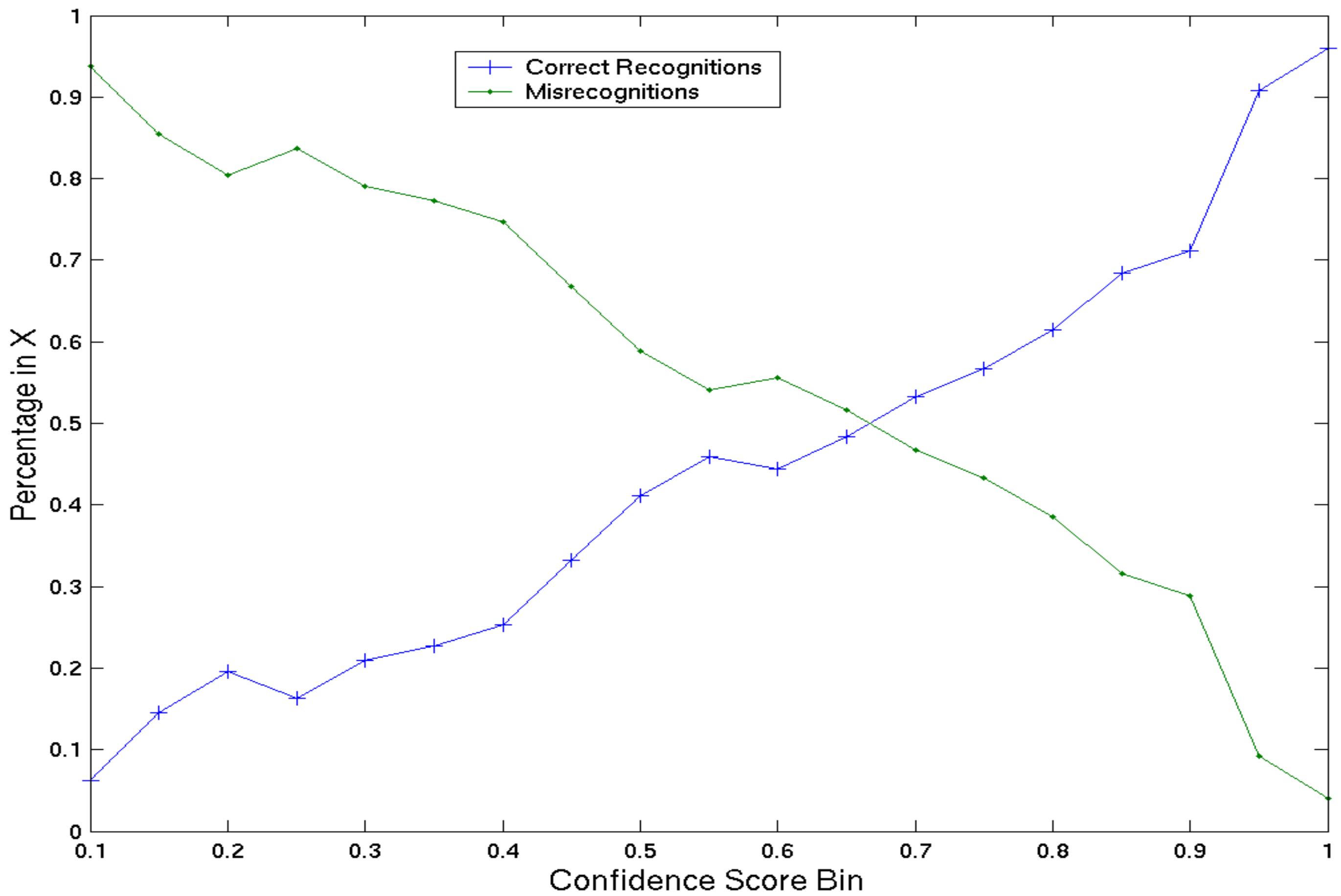
$$FR = \frac{\text{\# of correctly recognized words that are rejected}}{\text{\# of words that are rejected}} \times 100\%$$

# False Acceptance vs. False Rejection

- ASR 1-best posteriors
- Augmented ASR 1-best posteriors (using word lattices)
- Pivot alignments using time
- Pivot alignments without time



# Percent Correct/Misrecognition



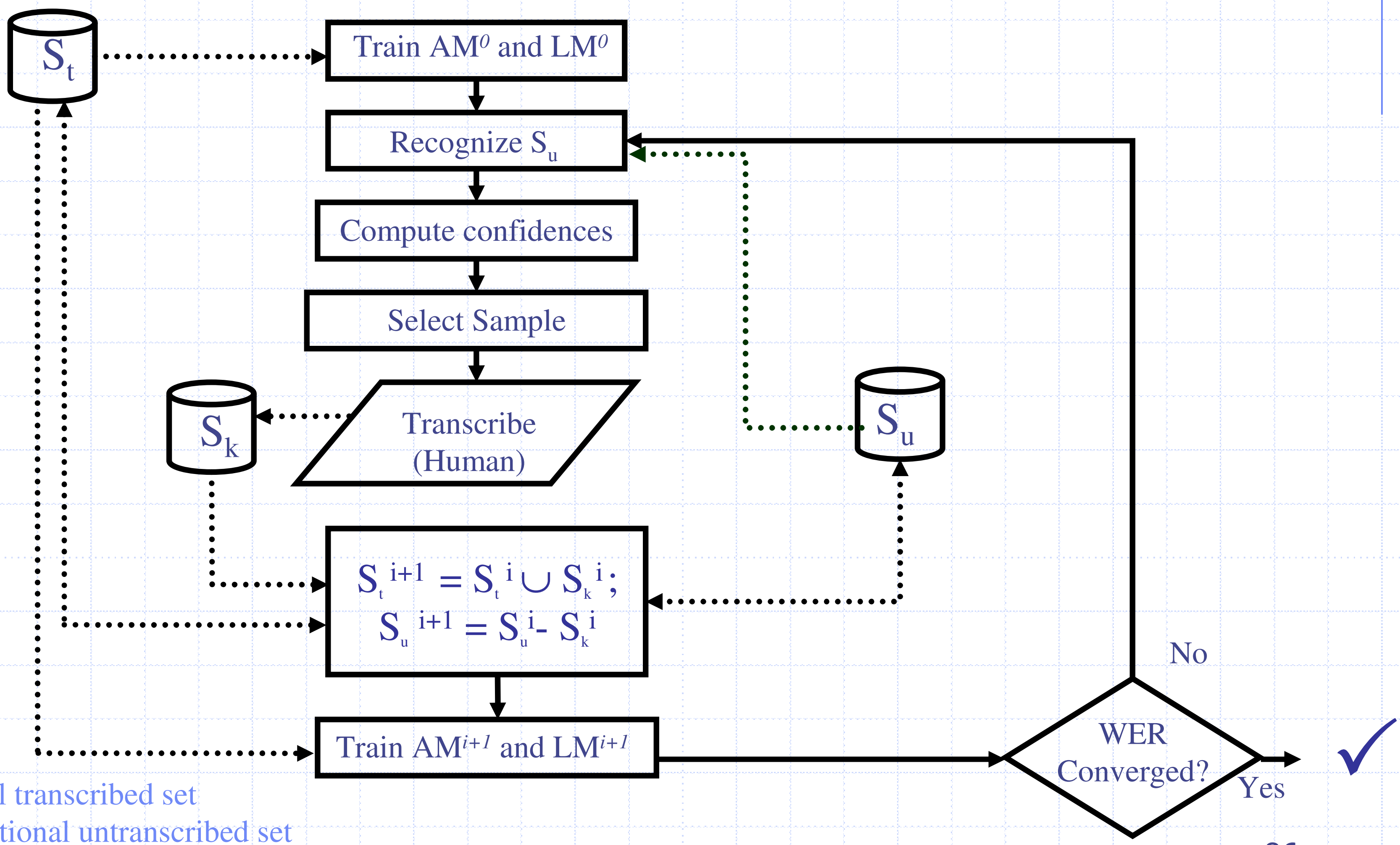
# Active Learning for Automatic Speech Recognition

- (Hakkani-Tür et al., ICASSP 2002)
- (Kamm, Ph.D. Thesis, 2004)

# Active Learning for ASR

- Goals:
  - Reduce the amount of transcribed data needed without reducing accuracy.
  - Optimize the performance using a given set of transcribed data.

# Algorithm




$S_t$: Initial transcribed set

$S_u$: Additional untranscribed set

$S_k$: Intermediate set to be transcribed

# Utterance Scores

- The algorithm is independent of the way utterance scores are computed, as long as they are good quality.
- We compute utterance scores, using word confidence scores. $U=w_1,\ldots,w_k$
  - Mean confidence score

$$c(U)=\frac{1}{k}\sum_{i=1}^{k}c(w_i)$$

  - Voting

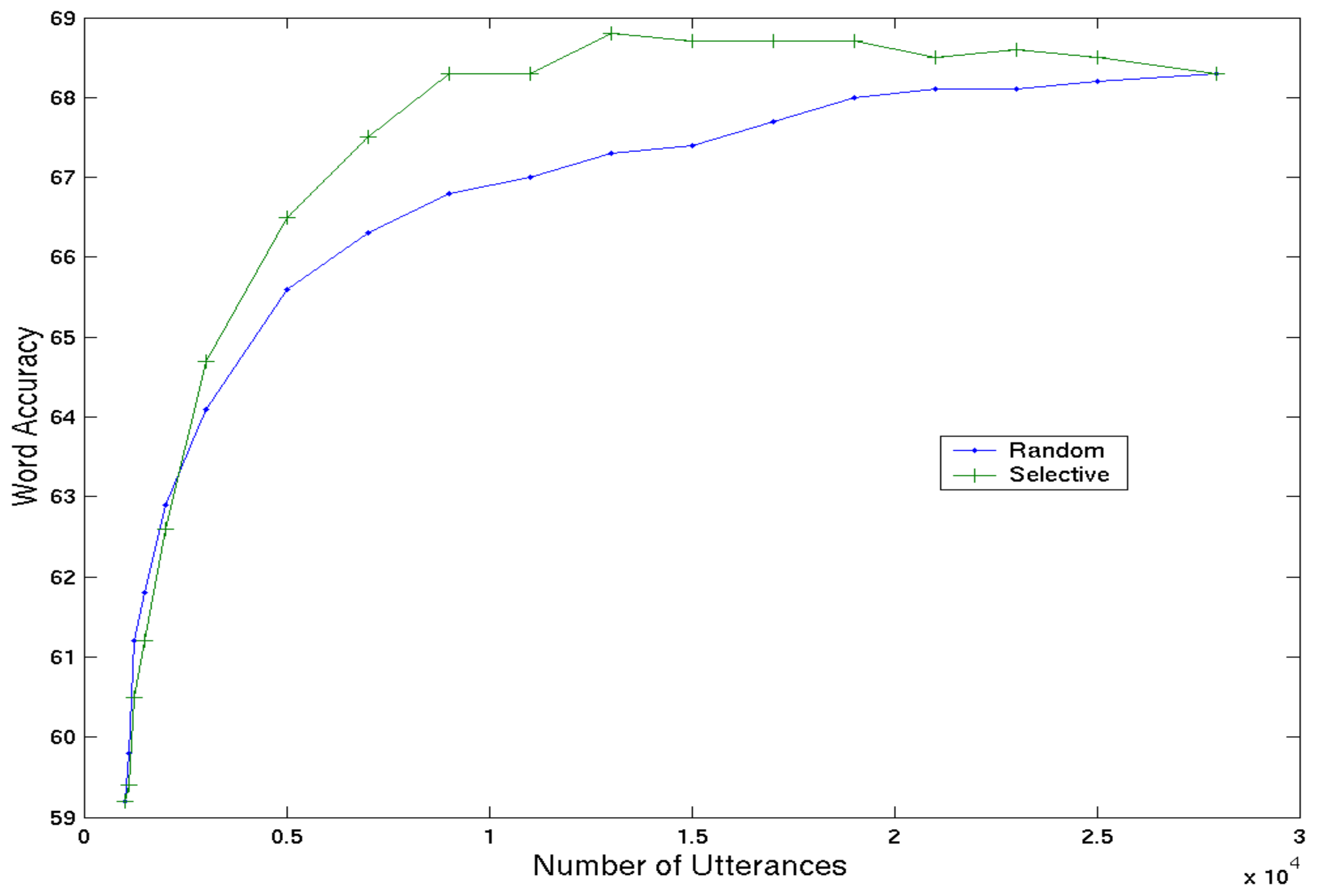
$$c(U)=\frac{1}{k}\sum_{i=1}^{k}f(c(w_i)) \text{ where } f(c(w_i))=\begin{cases}1, & c(w_i)>\text{threshold}\\ 0, & \text{otherwise}\end{cases}$$

# Active Learning Expt(1)

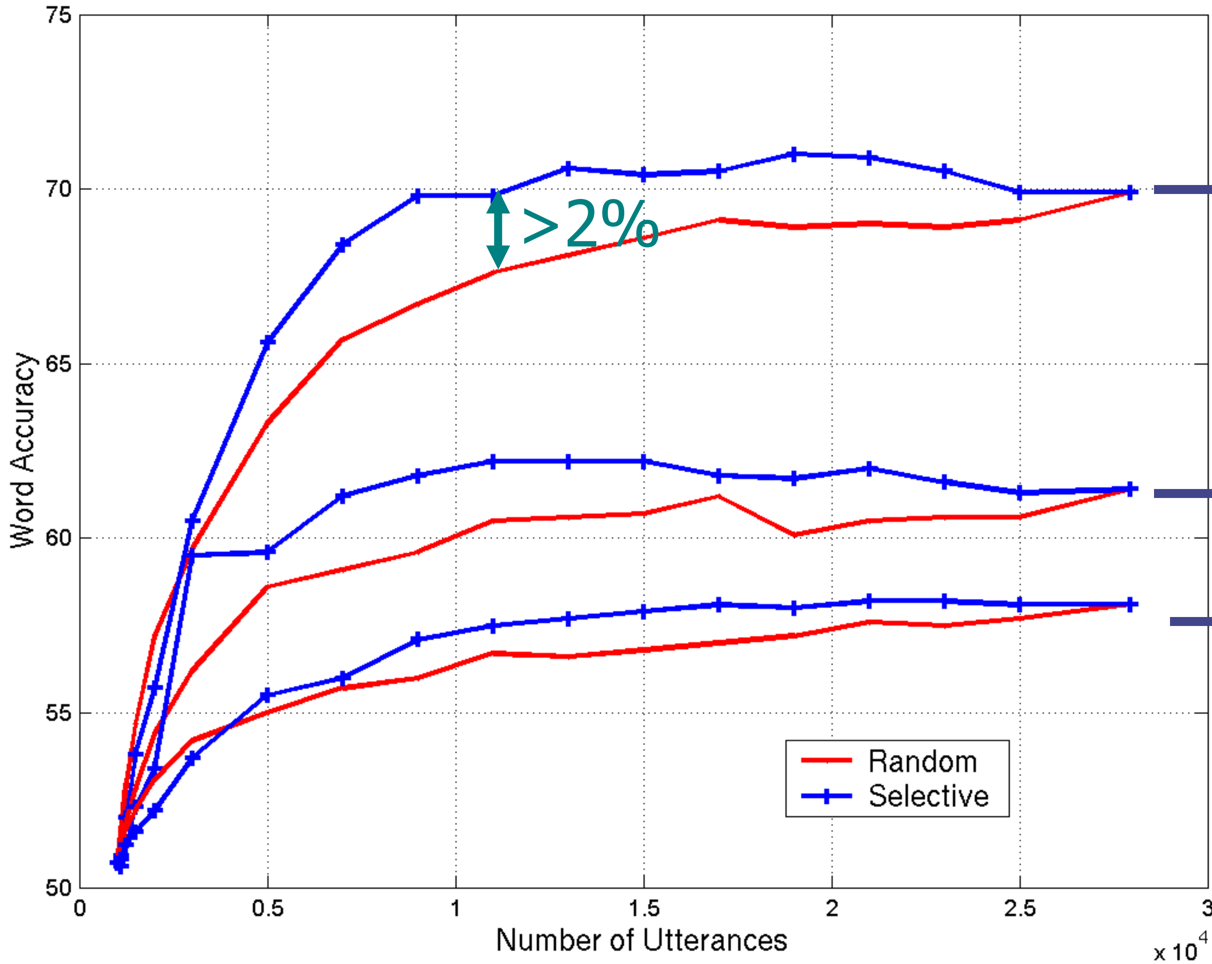
- Data collected from HMIHY?$^{SM}$ field trials
  - ~100K utterances
- All utterance turns (80 prompts)
- Bootstrap data for LM and scoring
  - HM data collection
- Data is pooled and sampled
- No time ordering constraint

# Active Learning Expt(1)

- Halve data size requirement for a given Φ
- Improve over asymptotic performance



# Active Learning Expt (2)



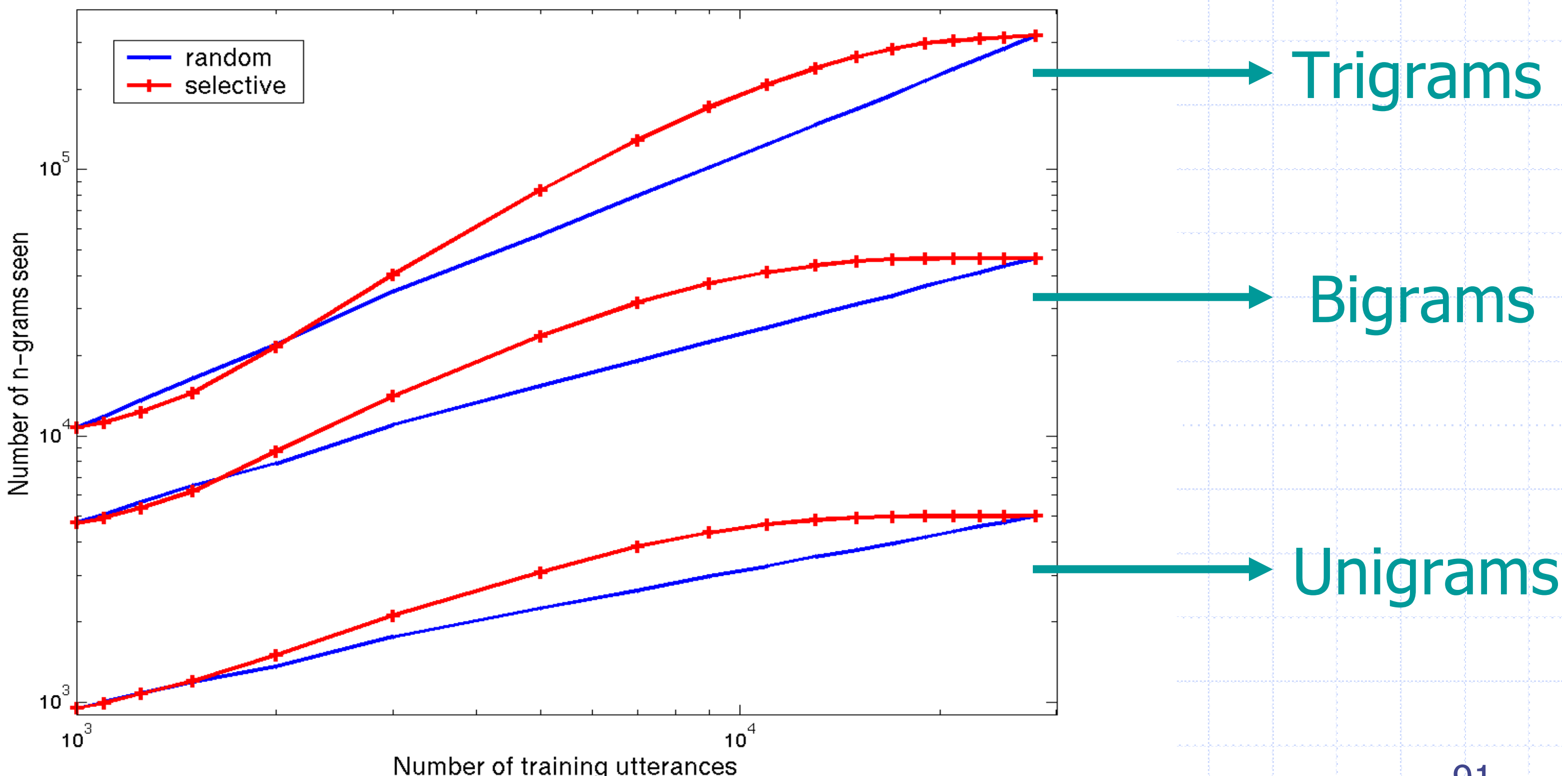

AM and LM are retrained

Only AM is retrained

Only LM is retrained

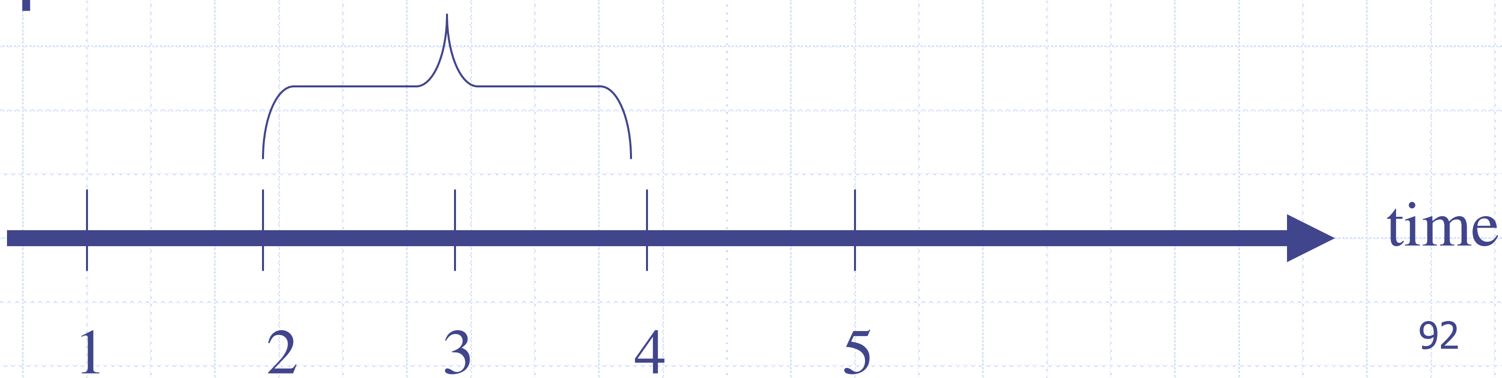
# Why does Active Learning work?

- Language modeling:
  - discover new words
  - discover new n-grams

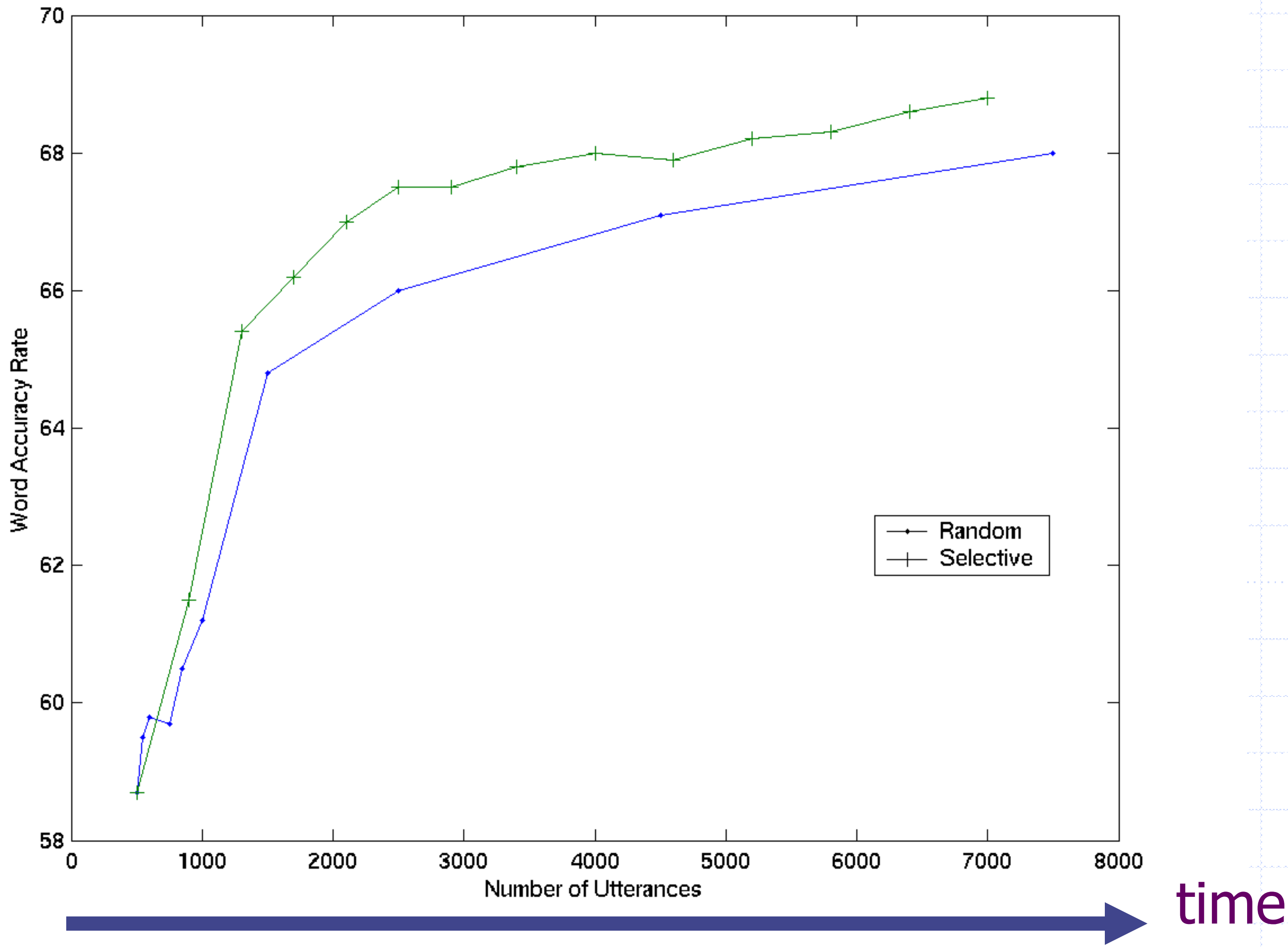


# Active Learning Expt(3)

- Data is time ordered and time-dependent data bin is used for selective sampling
- Time window for selective sampling
- Data is "forgotten" after n days
- Experiment close to operation modus operandi



# Active Learning Expt(3)

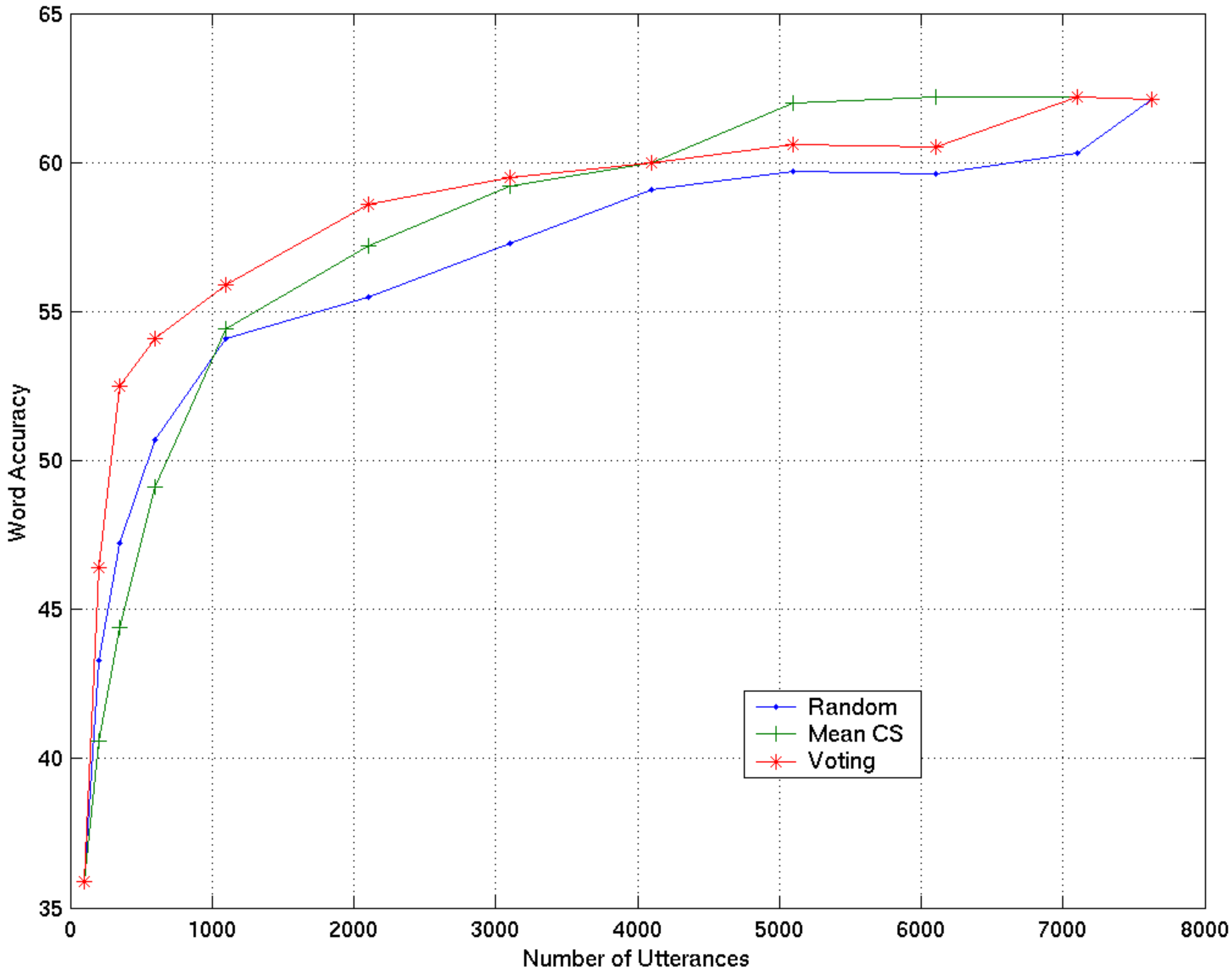


# Active Learning Expt(1)

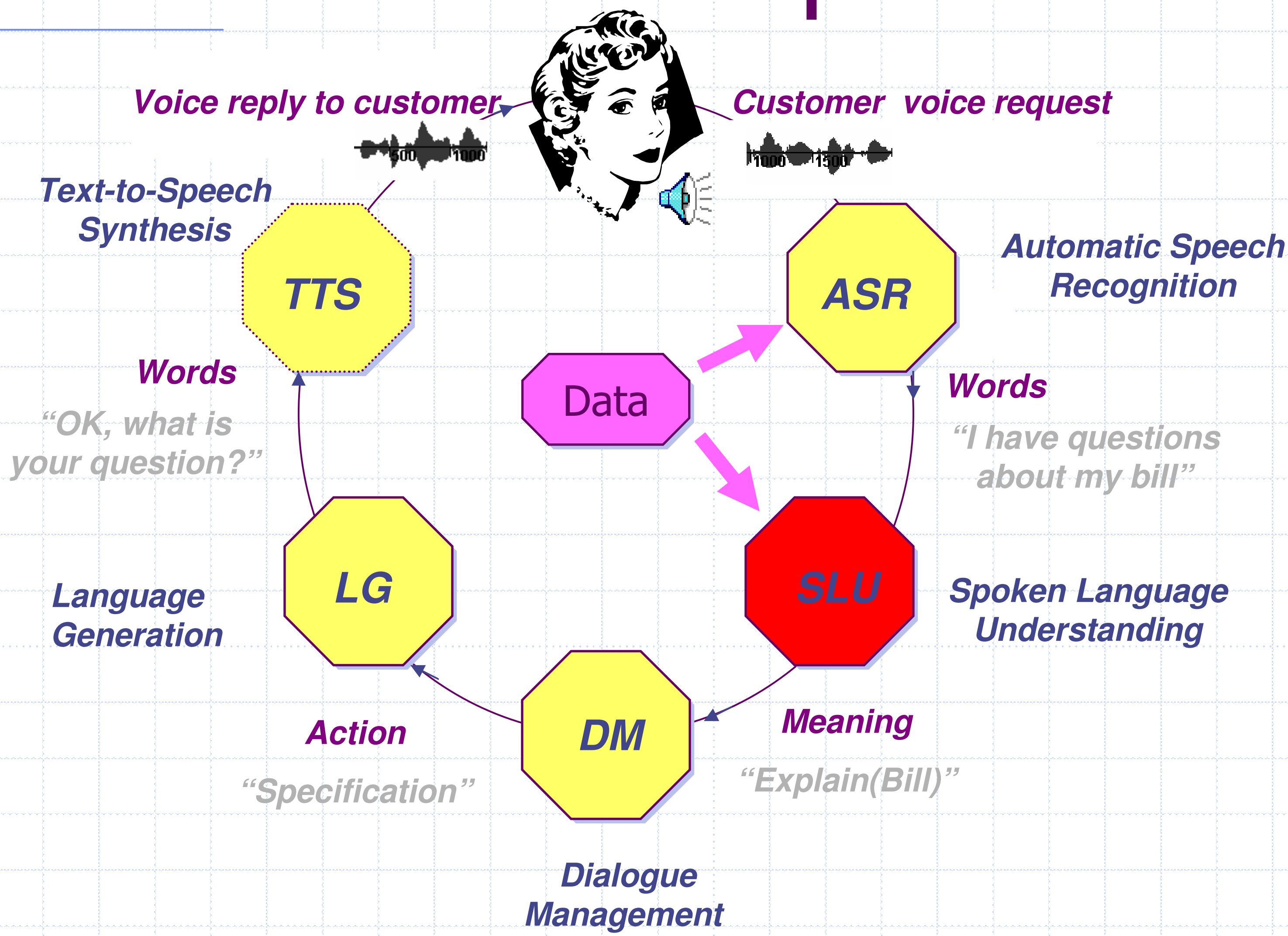
- Data collected from TTS Help Desk Trial
  - 8K utterances
  - Average length 5 words
  - Channel distortions (not matched AM)
- All utterance turns
- Bootstrap data for LM and scoring
  - Web-Mail data
- Data is pooled and sampled
- No time ordering constraint

# Active Learning Expt(2)

(TTS Help Desk)



# Human-Machine Spoken Dialog

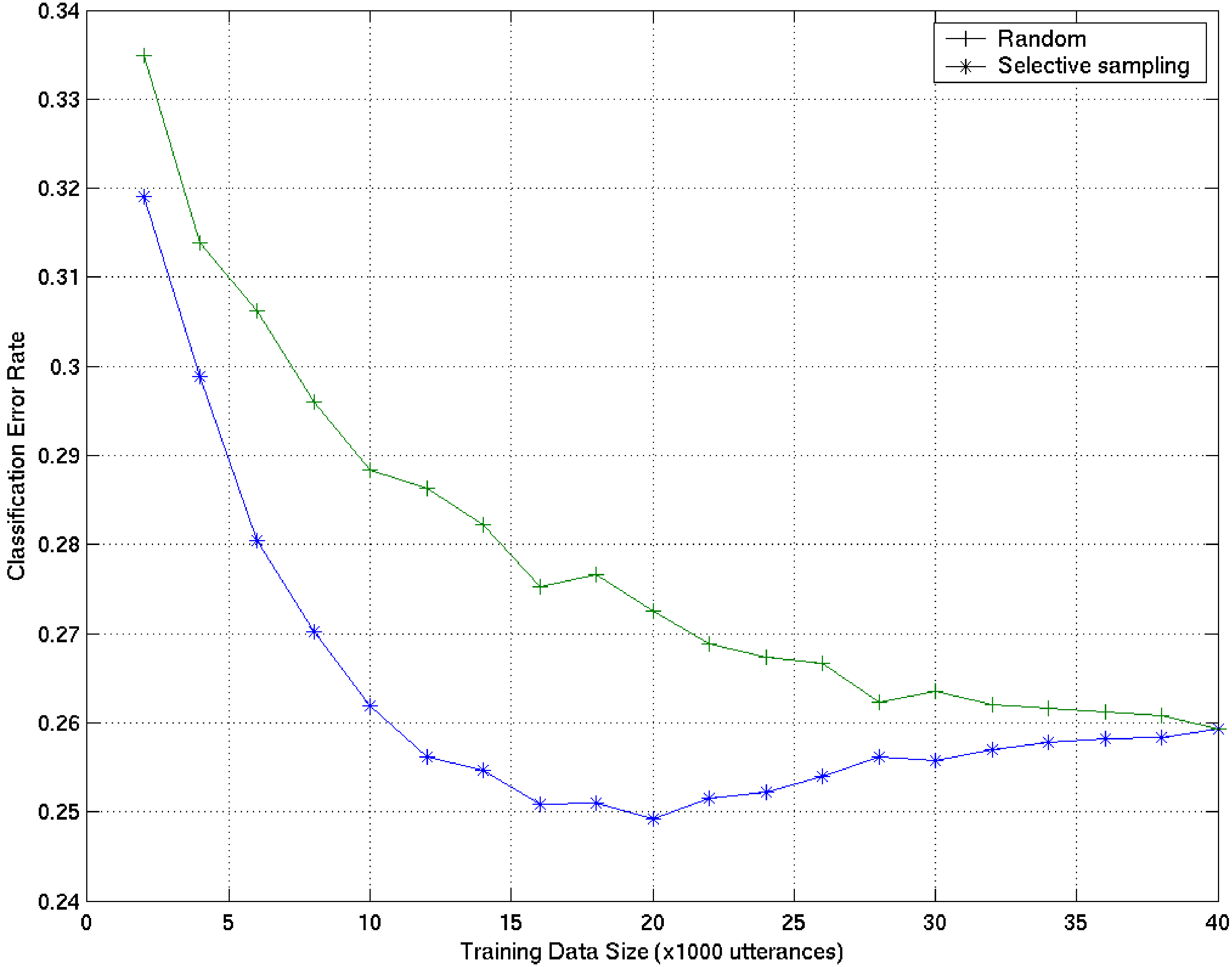


# Understanding User Intent

- **Greeting Prompt:** AT&T … How may I help you?
- **User:** I have questions about my bill
  - **Call-type**: *Explain(Bill)*
- **Specification Prompt:** OK, what is your question?
- **User**: I have a couple of numbers I wanna check out
  - **Call-type**: *Explain(Bill_UnrecognizedNumber)*
- **Confirmation Prompt:** Would you like to look up a number you don't recognize on your bill?
- **User:** Several of them
  - **Call-type**: *Yes*

# Call Classification

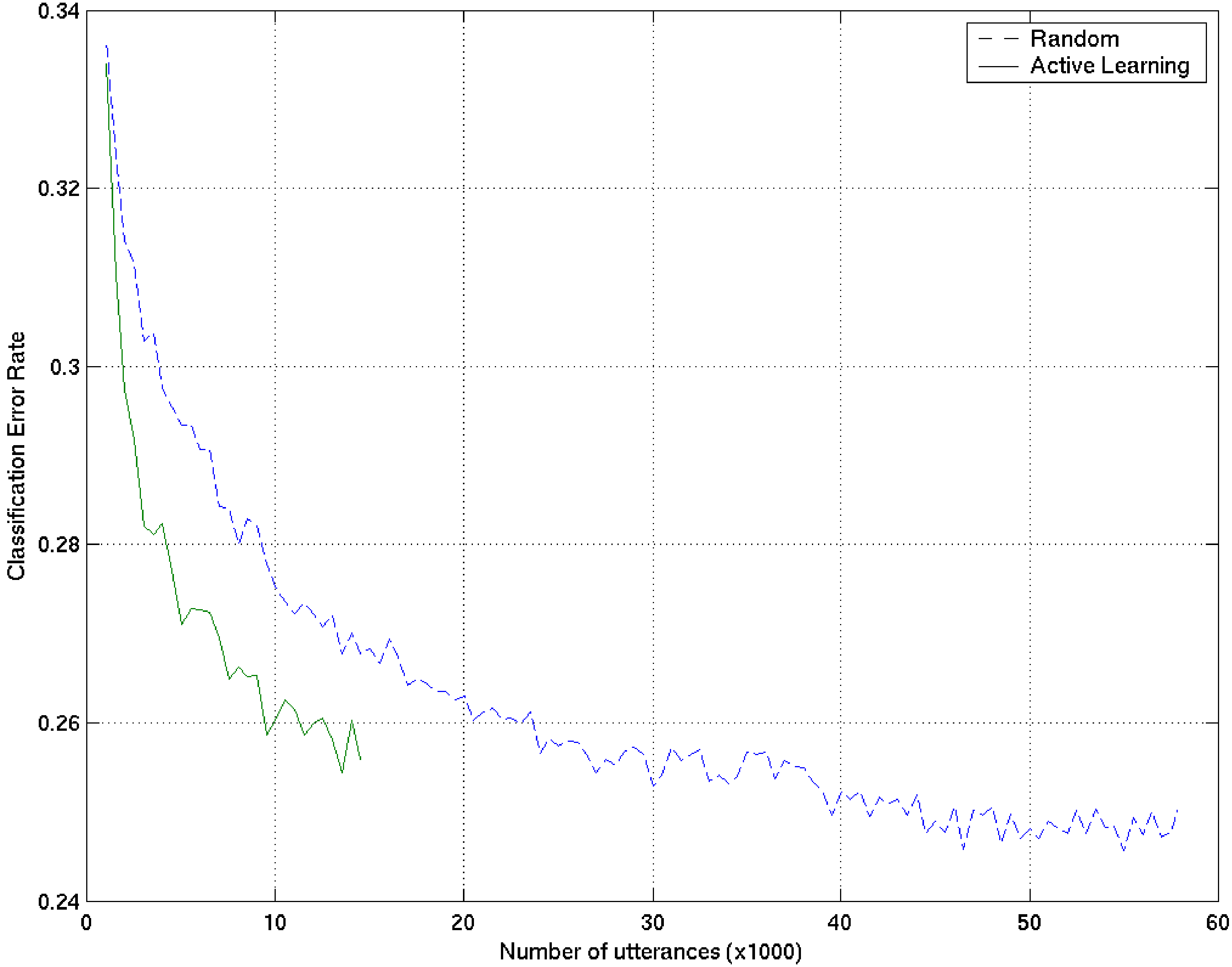
- *Tur, Schapire, and Hakkani-Tür, ICASSP'03*
- 56 call types in total (0300)
- Classifier: Boosting
- Fixed pool



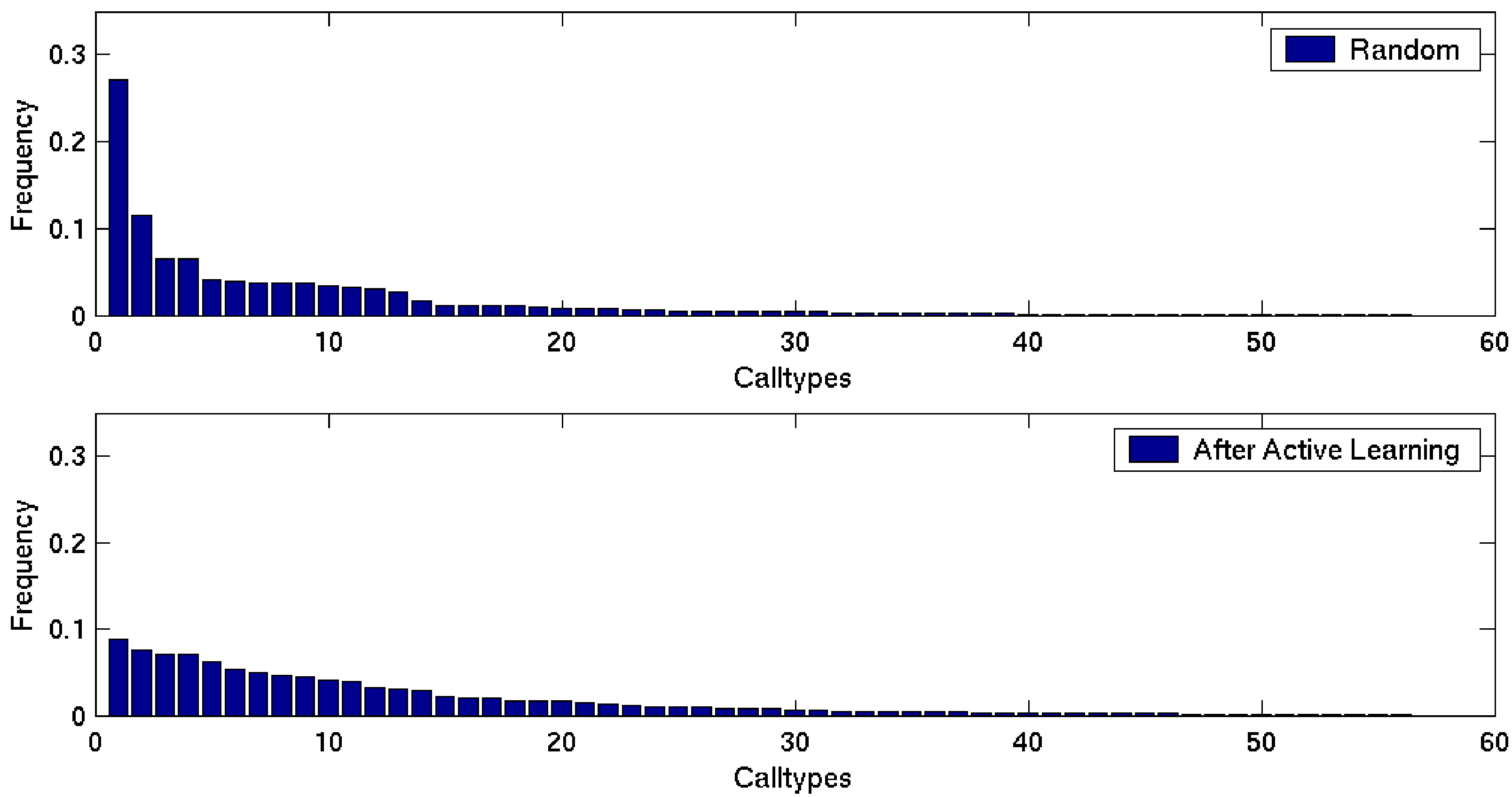
Random
Selective sampling
Classification Error Rate
Training Data Size (x1000 utterances)
0.34
0.33
0.32
0.31
0.3
0.29
0.28
0.27
0.26
0.25
0.24
0
5
10
15
20
25
30
35
40

# Call Classification

- *Tur, Hakkani-Tür, and Schapire; ICASSP 2003*
- 56 call types in total (0300)
- Classifier: Boosting
- Dynamic Pool (1/4 of the candidate utterances selected at each iteration)


Random
Active Learning
Classification Error Rate
0.34
0.32
0.3
0.28
0.26
0.24
0
10
20
30
40
50
60
Number of utterances (x1000)

# Unbalanced Data Problem



# Unbalanced Data Problem

- Active learning changes the prior probabilities significantly.
- Halved the data from 10K to 5K by ignoring the utterances with calltypes occurring more frequent than a certain threshold.

| *Training Set* | *Test Set Classification Error Rate* |
|---|---|
| Random 5K | 29.12% |
| Biased 5K | 30.81% |

- Biasing distributions hurt the performance!

# One Solution

- This is not a paradox. If we can find a solution to this problem, active learning may perform better.
- *Lewis and Catlett, ICML'94* suggested:
  - Changing priors while training
  - Making false-positives more costly than false-negatives (C4.5 supports this)

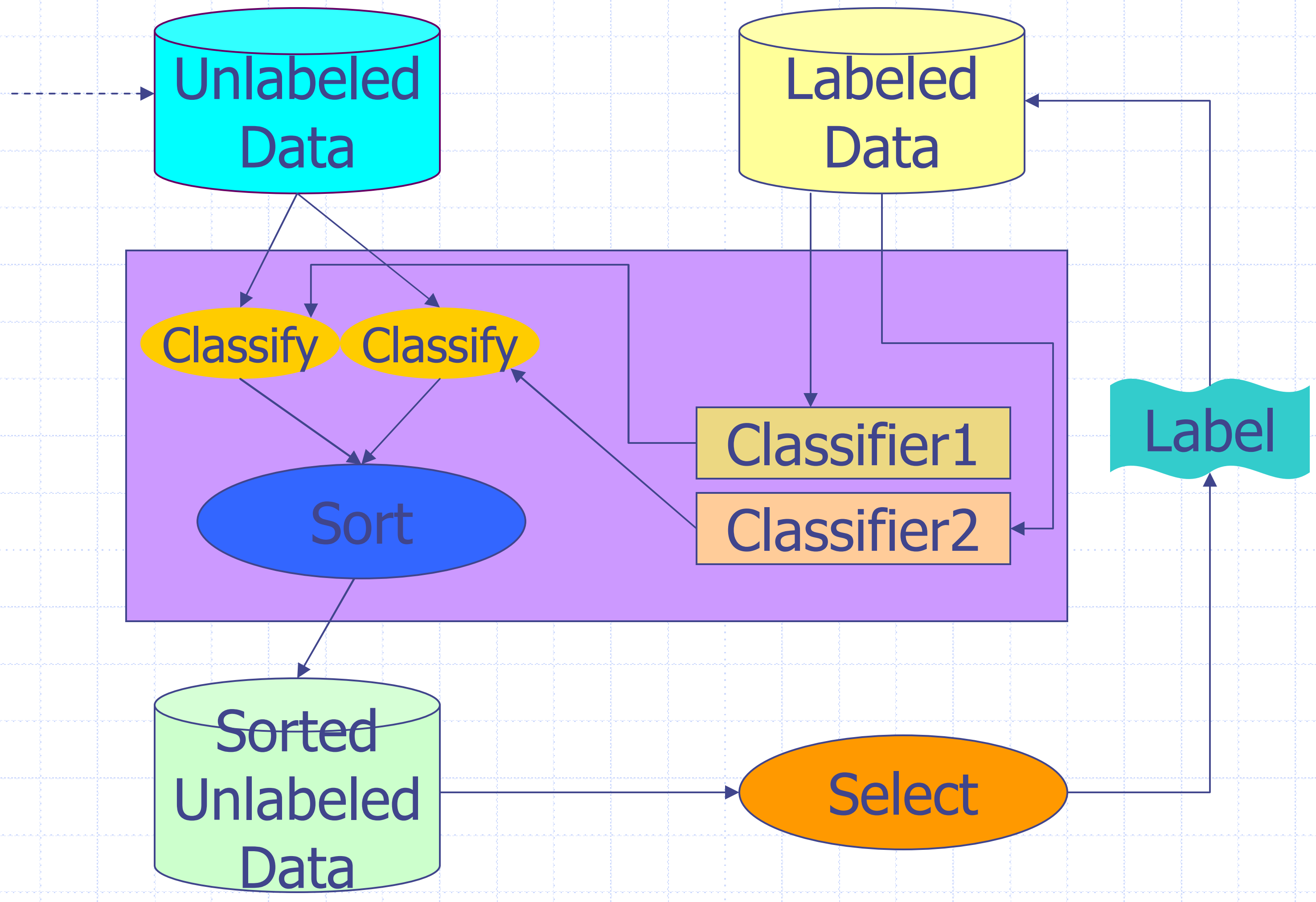
# Outline

- Algorithm Dimension:
  - Passive vs. Adaptive Learning
  - Active Learning
    - Certainty-based
    - Committee-based
  - Unsupervised Learning
  - Combining Active and Unsupervised Learning

# Committee-based Active Learning

- Train multiple classifiers using initial training data
- While (labelers/data available) do
  - Label the data in the pool using all classifiers
  - Sort them according to **disagreement** between classifiers
  - Select the top $k$ of them.
  - Label and add the selected ones to the training data
  - Re-train the classifier
  - Update the pool

# Committee-Based Active Learning



# Selected Bibliography for Committee-Based Active Learning

- Seung, Opper, Sompolinsky COLT'92
- Freund, Seung, Shamir, Tishby ML'97
- Liere and Tadepalli AAAI'97 (Text Categorization)
- Engelson and Dagan JAIR'99 (POS Tagging)
- Tur, Schapire, and Hakkani-Tür ICASSP'03 (Call Classification)
- Osborne and Baldridge, EMNLP'03, NAACL'04 (Parsing)

# Part of Speech Tagging

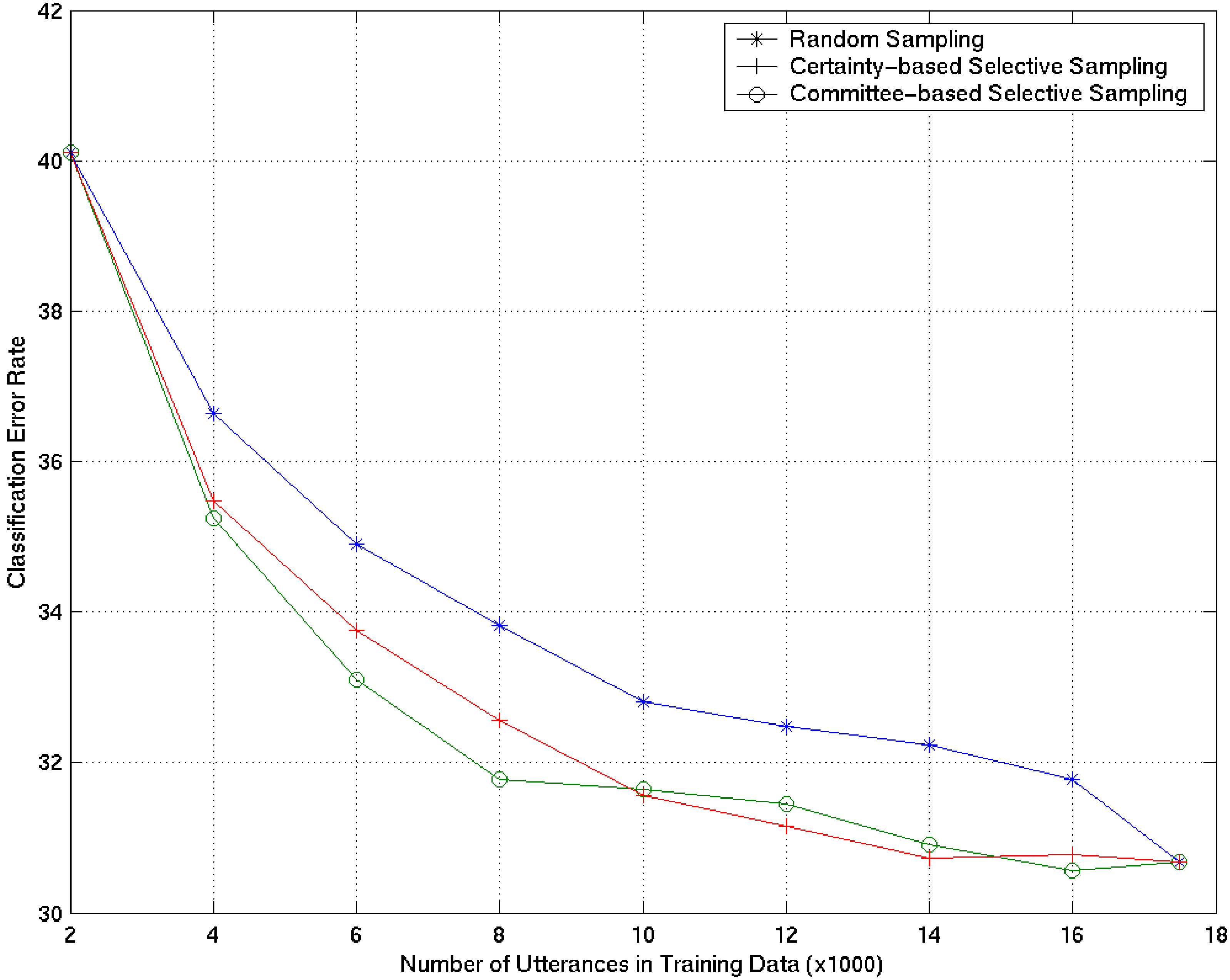
- *Engelson and Dagan JAIR'99*
- Part-of-speech tagging using HMMs
- Degree of disagreement for sample $w$: normalized entropy of committee classifications

$$D(w) = -\frac{1}{\log \min(k, |C|)} \sum_c \frac{V(c,w)}{k} \log \frac{V(c,w)}{k}$$

- Reduced the amount of human-labeled data needed by a factor of 4 using 10 committee members.

# Call Classification

- *Tur, Schapire, and Hakkani-Tür, ICASSP'03*
- 56 call types in total
- Fixed pool
- 2 committee members using 2 different classifiers: SVM and Boosting


Random Sampling
Certainty-based Selective Sampling
Committee-based Selective Sampling
Classification Error Rate
Number of Utterances in Training Data (x1000)
42
40
38
36
34
32
30
2
4
6
8
10
12
14
16
18

# Parsing (HPSG)

- (Osborne and Baldridge, EMNLP'03, NAACL'04)
- A committee of parsers is trained using different and independent feature sets:
  - Configurational (e.g. parent, grandparent, sibling relationships)
  - *N*-gram (*n*-grams over tree nodes)
  - Conglomerate (features from phrase structures)
- Cost of manual annotation is not equal to the number of utterances hand-labeled, but is proportional to the number of disambiguation decisions the labelers have to make.
- 73% reduction in the cost of annotation.

# Outline

- Algorithm Dimension:
  - Passive vs. Adaptive Learning
  - Active Learning
    - Certainty-based
    - Committee-based
  - Unsupervised Learning
  - Combining Active and Unsupervised Learning

# Unsupervised Learning

- **Goal**: to exploit the unlabeled utterances
  - to train better models
  - to train in a shorter time frame
  - to adapt fast to changes

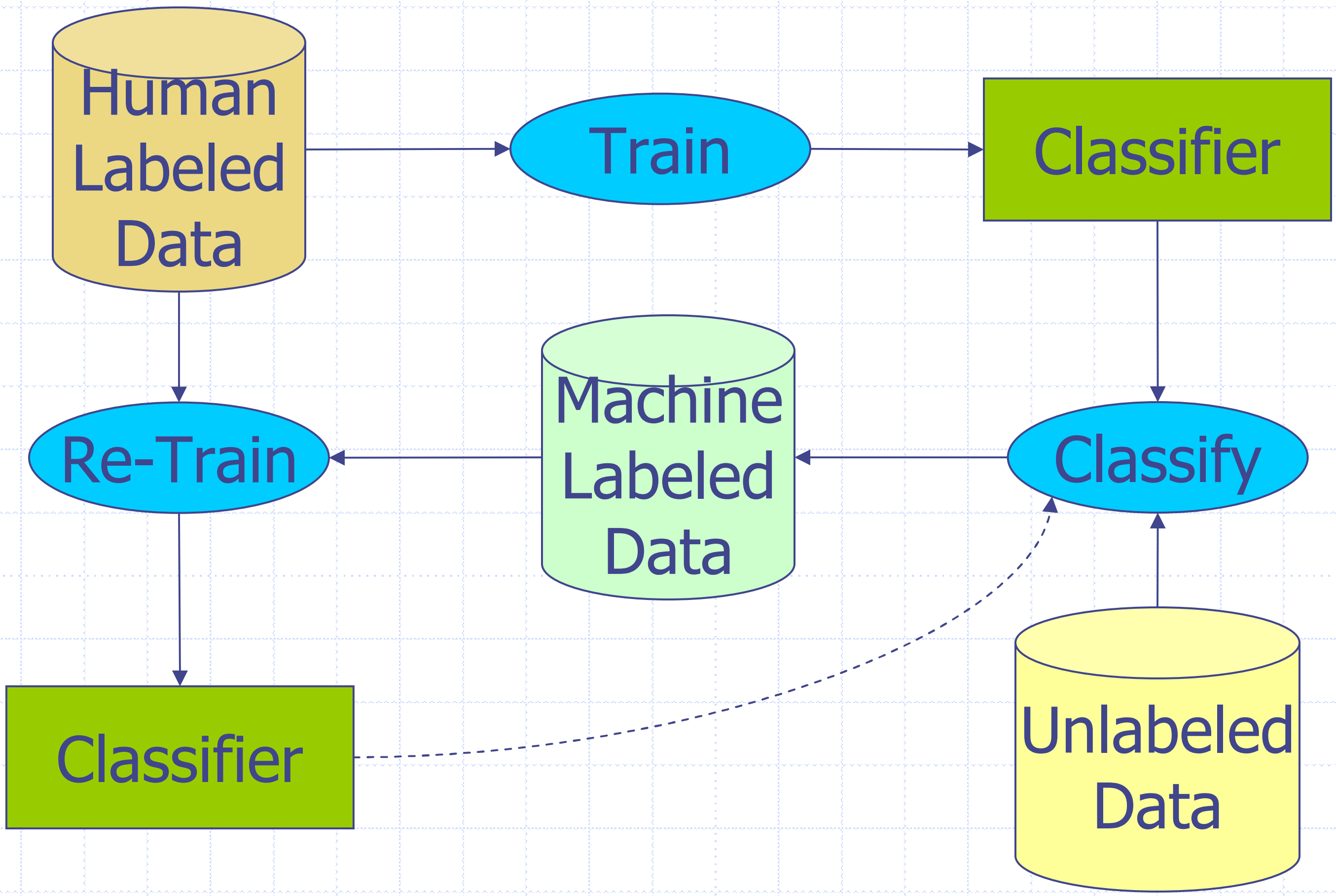
# Selected Bibliography for Unsupervised Learning


- Blum and Mitchell, COLT'98
- Nigam and Ghani, ICML'98
- Joachims, ICML'99
- Nigam, McCallum, Thron, and Mitchell, ML'00
- Nigam and Ghani, CIKM'00
- Ghani, ICML'02
- Tur and Hakkani-Tür, ES'03
- …

# Using EM

- *Nigam, McCallum, Thron, and Mitchell, ML'00*
- Train a classifier using human-labeled data (call this prior model: $\Pi$)
- Add unlabeled utterances:
  - Classify the unlabeled utterances with $\Pi$ (**Estimation**)
  - Add this machine-labeled data to the human-labeled data in a weighted manner and re-train the classifier (**Maximization**)
  - Iterate until model parameters converges
- 3-fold reduction in labeled data needed

# Unsupervised Learning



# Co-Training

- *Blum and Mitchell, COLT'98*
- Assume there are multiple views for classification

  e.g. Task: Web-page classification
  1. Words in the web-page
  2. Words in the hyperlinks pointing to that web page

1. Train multiple models using each view
2. Classify unlabeled data
3. Enlarge training set of the other using each classifier's predictions
4. Goto Step 1

- Halved the classification error rate
- Nigam and Ghani later extended this to Co-EM so that it uses probabilistic labels (*CIKM'00*)

# Unsupervised Learning for ASR

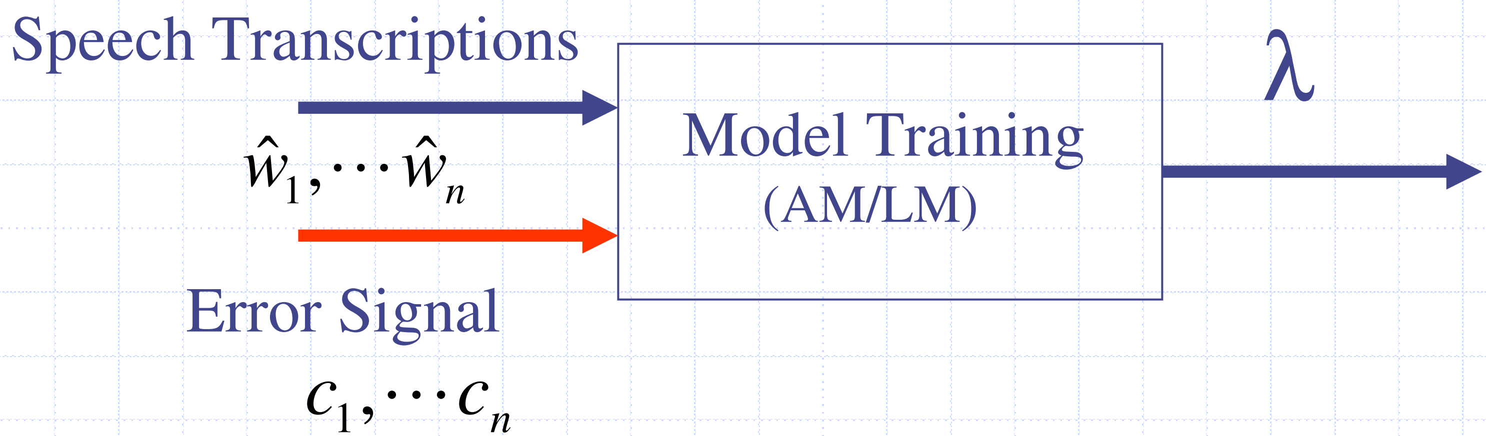
- Goal: Exploit untranscribed data to improve performance.
- Use of the error signal to exploit the untranscribed data.
- Use of extra information, such as TV captions.
- Combining active and unsupervised learning.

# Previous Approaches

- AM
  - TV captions (Kemp and Waibel, 1998, 1999).
  - Accurate portions of the ASR output (Zavaliagkos and Colthurst, 1998).
  - ASR output (Lamel et al., 2002).
- LM
  - Word confidence scores to extract the portions that are recognized correctly (Gretter and Riccardi, 2001).
  - ASR output (Stolcke, 2002).
  - ASR word lattices with posteriors (Roark and Bacchiani, 2003).
- Riccardi and Hakkani-Tür (Eurospeech, 2003).

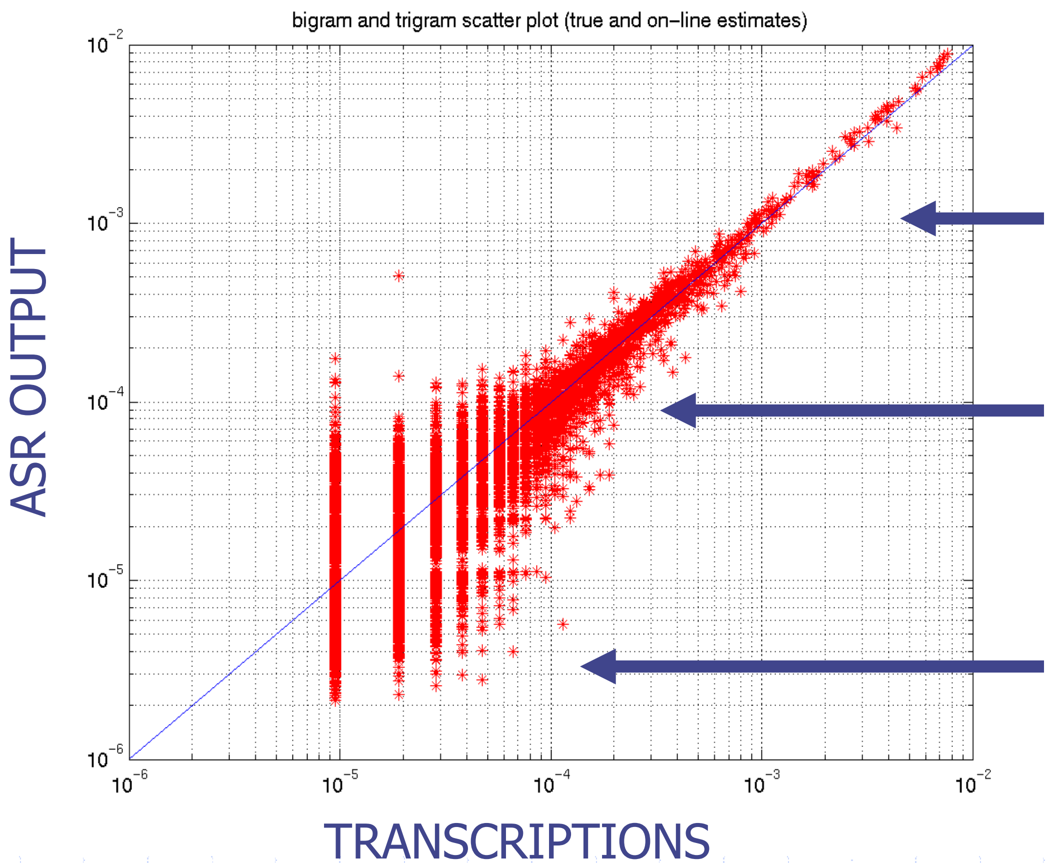
# Unsupervised Learning

$$C(w_i, w_{i+1}, w_{i+2}) = F(C(\hat{w}_i, \hat{w}_{i+1}, \hat{w}_{i+2}), \mathrm{c})$$



# Unsupervised Learning for ASR

- Estimate probabilities from ASR output.




**Highly Frequent** events

**Frequent** events

**Rare** events

# Results on 0300 Data

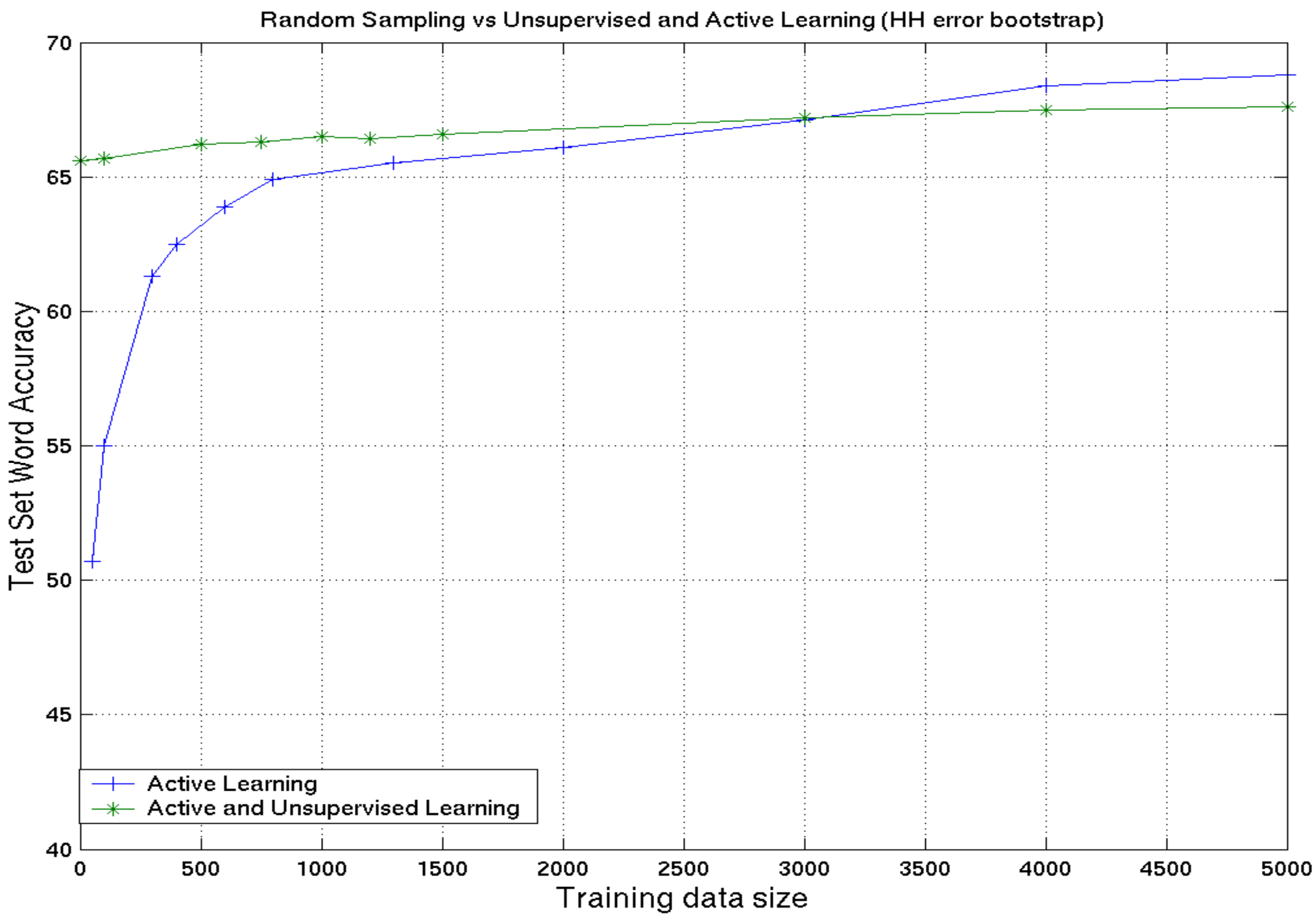
- Initial Set: random 1K H-M utterances (11K words)
- Additional Set: 27K H-M utterances
- Test Set: 1000 H-M utterances (~11K words)

| Training Set | Word Accuracy |
|---|---|
| Initial Set | 59.1% |
| ASR output of Additional Set | 61.5% |
| ASR output of Additional Set, with confidence scores | 62.1% |

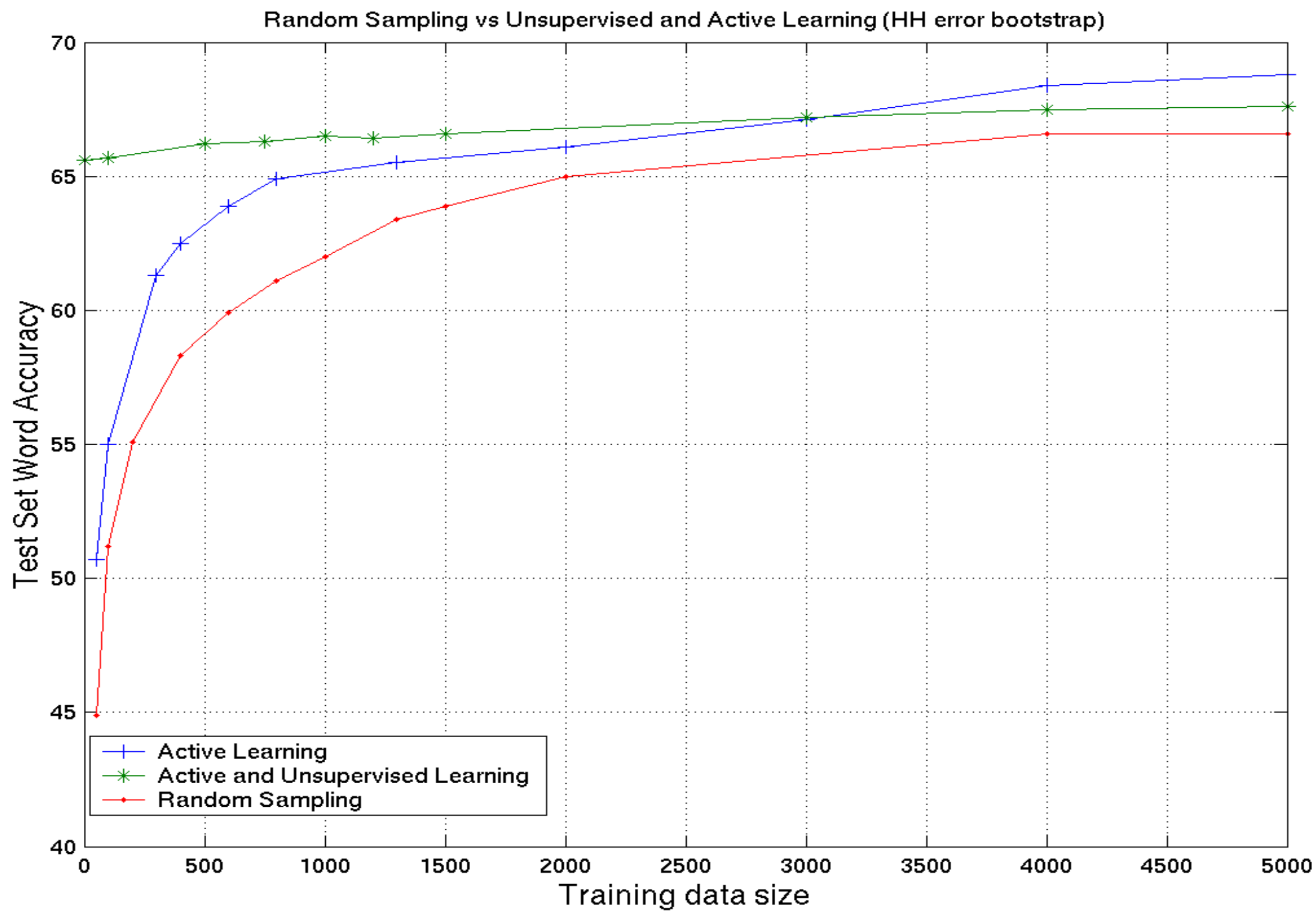
# Experiments with 0300 Data

- Initial Set: 8K H-H utterances
- Additional Set: 28K H-M utterances (~320K words)
- Test Set: 1000 H-M utterances (~11K words)

# Results on 0300 Data



# Results on 0300 Data



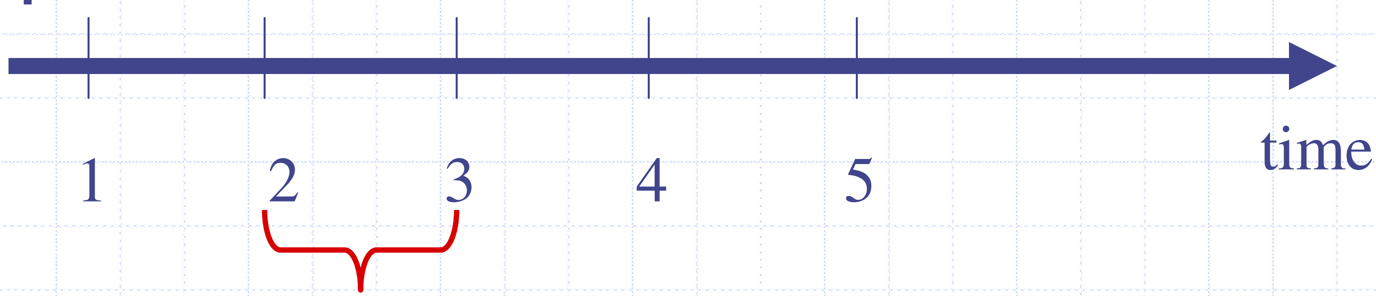
# Results on TTS Help Desk Data

- Initial Set: Web and e-mail data (~40 K words)
- Additional Set: 7,629 H-M utterances (~33K words)
- Test Set: 2,160 H-M utterances (~9.2K words)

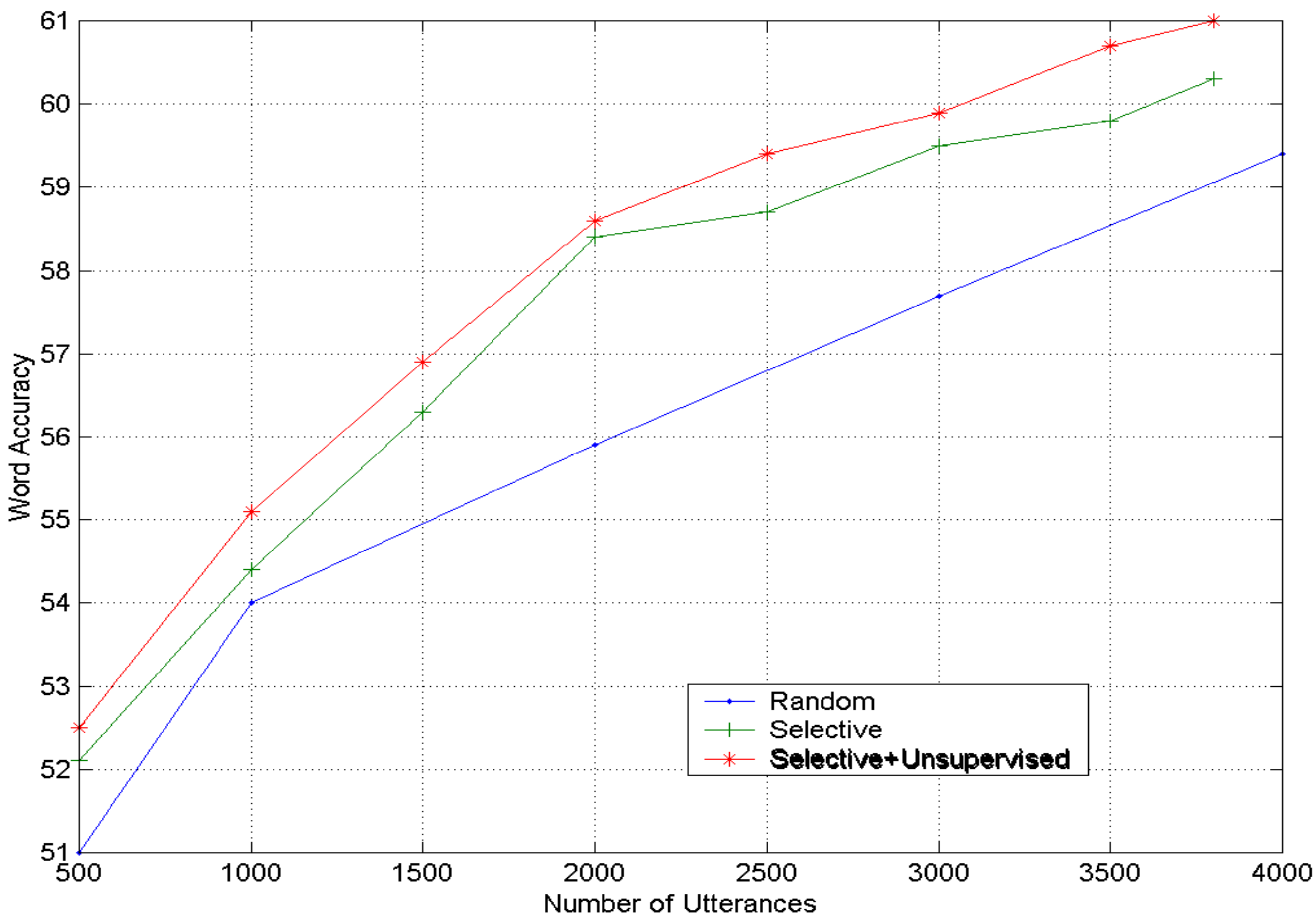
| **Training Set** | **Word Accuracy** |
|---|---|
| Initial Set | 42.2% |
| Initial Set + ASR output of Additional Set | 50.6% |
| Initial Set + Additional Set | 61.8% |

# Results on TTS Help Desk Data

- Data is time ordered and time-dependent data bin is used for selective sampling
- Time window for selective sampling
- Data is only used for unsupervised learning after n days.
- Experiment close to operation modus operandi



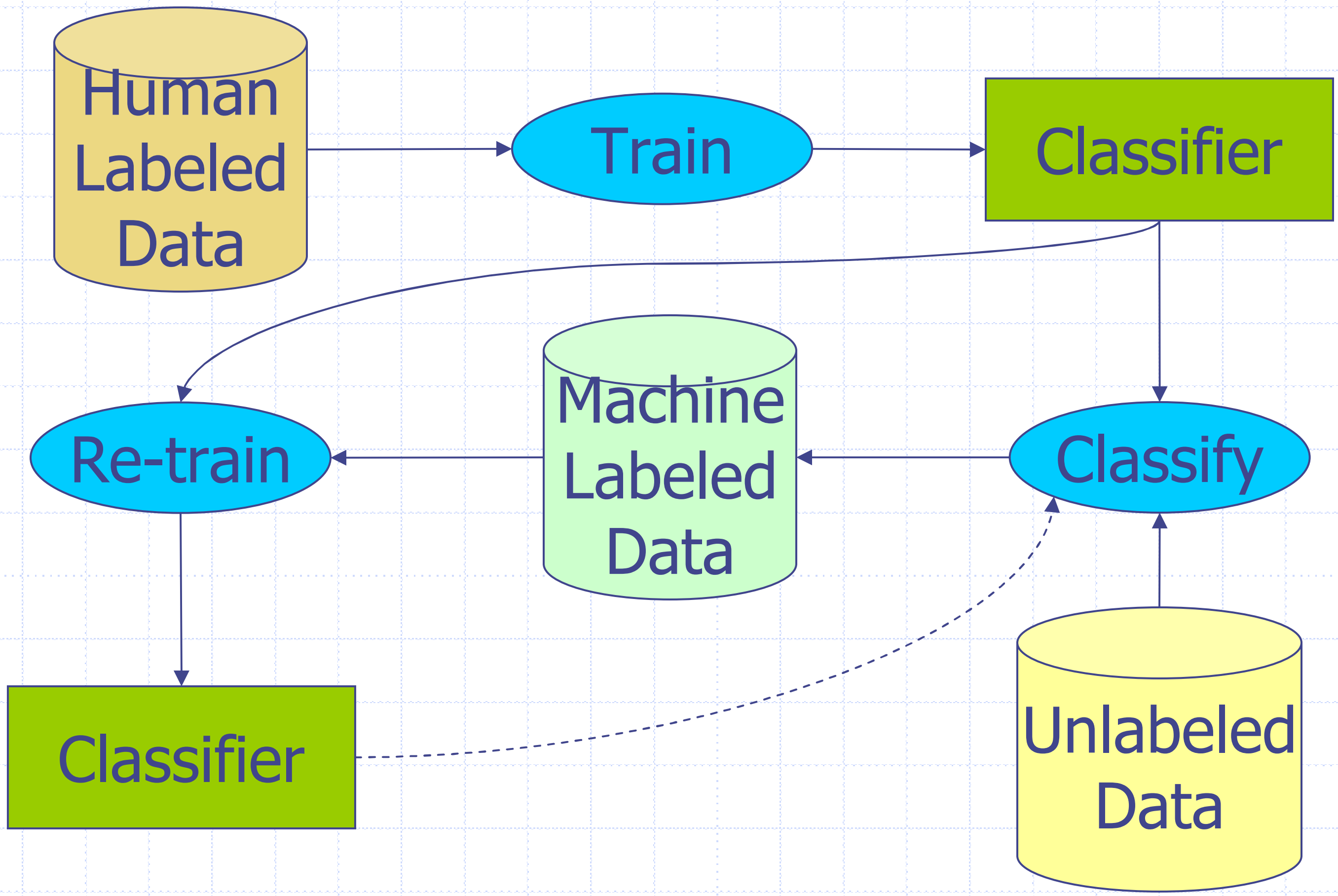
# Results on TTS Help Desk Data



# Unsupervised Learning in Boosting

- *Tur and Hakkani-Tür, Eurospeech'03*
- Train the Boosting classifier using human-labeled data (call this prior model: $\Pi$)
- Augment $\Pi$ with unlabeled utterances
  - Classify the unlabeled utterances with $\Pi$
  - Use the top calltype or calltypes exceeding some threshold as the label of that utterance
  - Augment the classifier using unlabeled data changing the loss function so that it fits both
    - the prior model, $\Pi$, and
    - the new unlabeled data

# Unsupervised Learning in Boosting



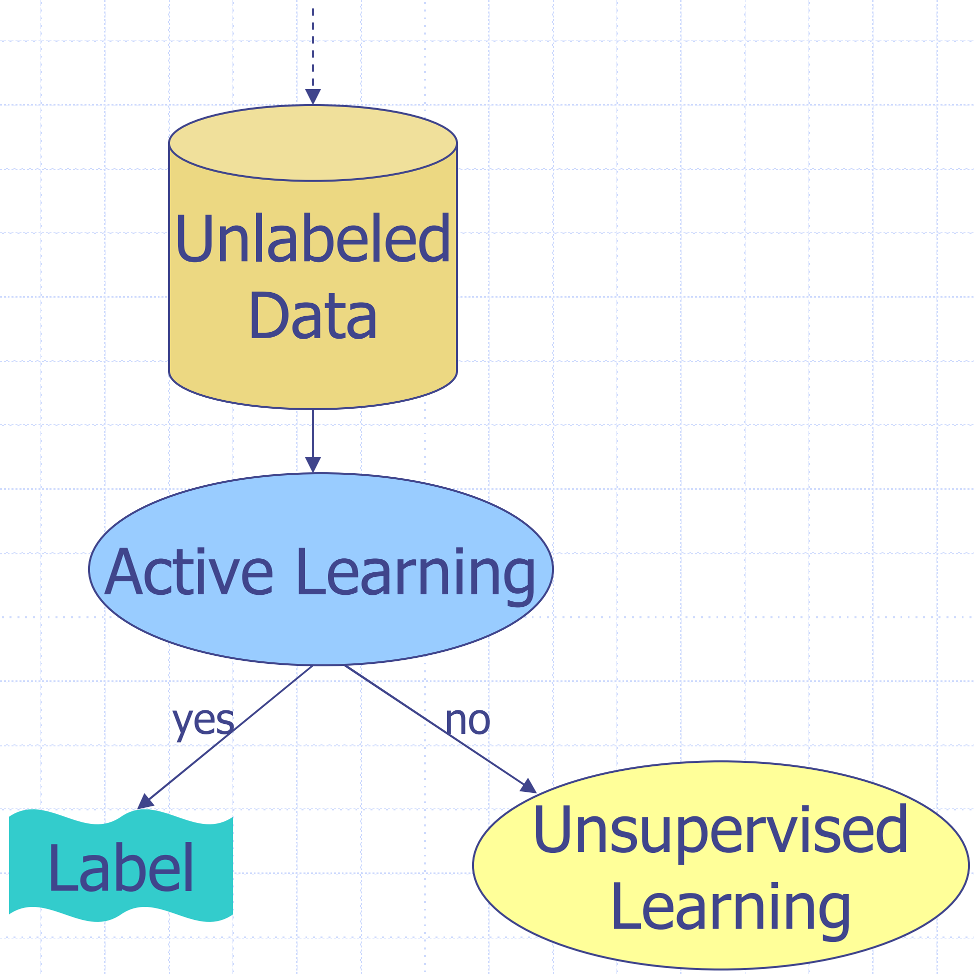
# Outline

- Algorithm Dimension:
  - Passive vs. Adaptive Learning
  - Active Learning
    - Certainty-based
    - Committee-based
  - Unsupervised Learning
  - Combining Active and Unsupervised Learning

# Combining Active and Unsupervised Learning

- Train a classifier using initial training data
- While (labelers/data available) do
  - Select $k$ samples for labeling using ***active learning***
  - Label and add these selected ones to the training data and re-train the classifier.
  - Exploit the unselected data using ***unsupervised learning***
  - Update the pool.

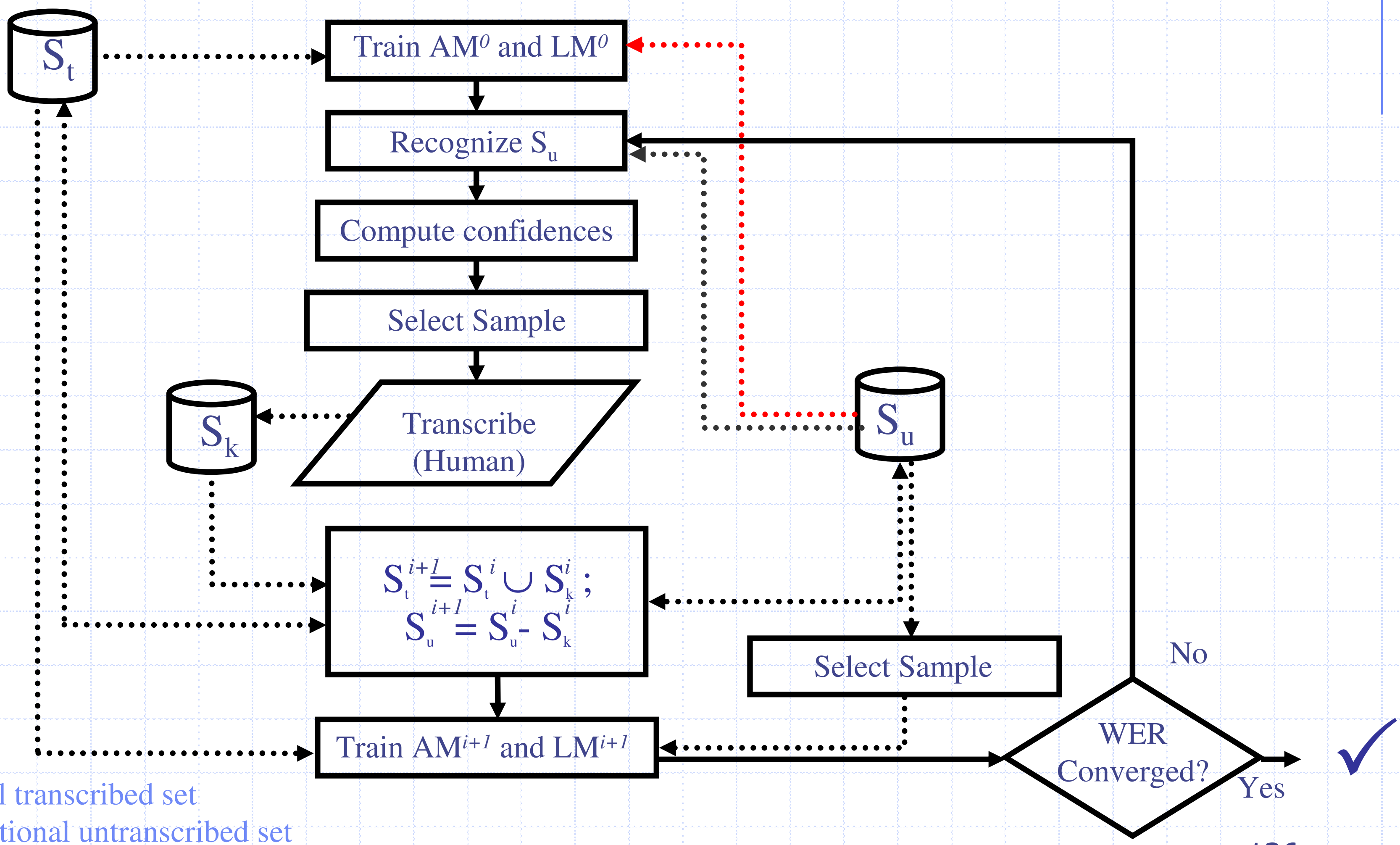
# Combining Active and Unsupervised Learning



# Selected Bibliography for Combining Active and Unsupervised Learning


- McCallum and Nigam, ICML'98
- Muslea, Minton, and Knoblock, ICML'02
- Tur, Hakkani-Tür, and Schapire, not appeared yet

# Active and Unsupervised Learning for ASR




$S_t$: Initial transcribed set
$S_u$: Additional untranscribed set
$S_k$: Intermediate set to be transcribed

# Exploiting Untranscribed Data

- $X$ is transcribed text, $x$ and $y$ are n-grams.

$$C(x) = \sum_{y \in X} \delta_x(y)$$

- $X$ is ASR output, where every n-gram $y$ has a confidence score, $c(y)$,

$$\begin{aligned} C_u(x) &= \sum_{y \in X} c(y) \times \delta_x(y) \\ &= \sum_{y \in X} (1 - e(y)) \times \delta_x(y) \\ &= C(x) - \sum_{y \in X} e(y) \times \delta_x(y) \end{aligned}$$

# N-gram Confidence Scores

- If we represent each n-gram $X$ as $x_1, \ldots, x_n$, the confidence score of each n-gram can be:

$$c(X) = \sqrt[n]{\prod_{i=1}^{n} c(x_i)}$$

$$c(X) = c(x_n)$$

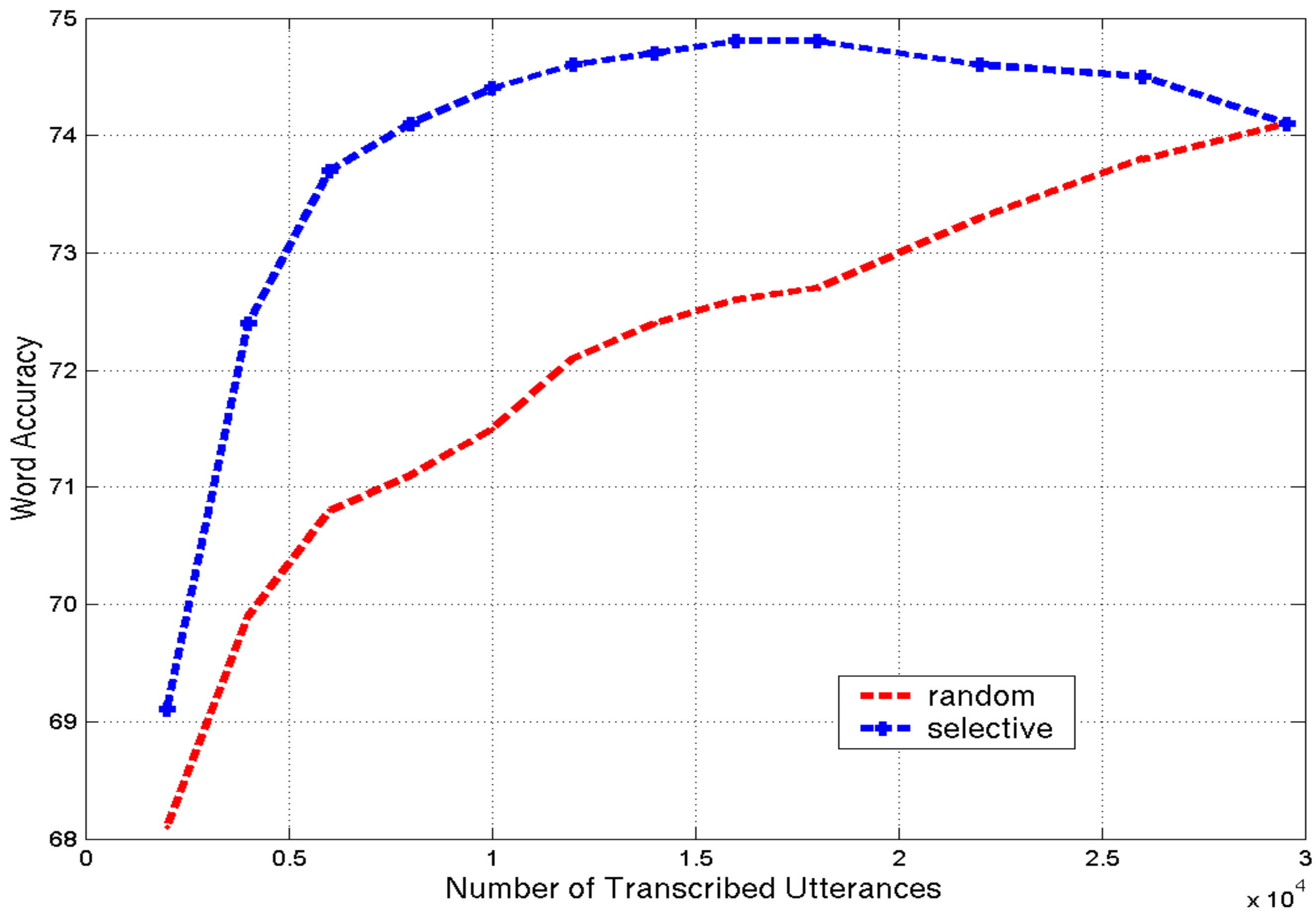
$$c(X) = \min_{x_i} c(x_i)$$

$$c(X) = \begin{cases} 1, \text{ if } \ \mathrm{c}(x_i) > \text{threshold}, & \forall x_i \\ 0, \text{ otherwise} \end{cases}$$

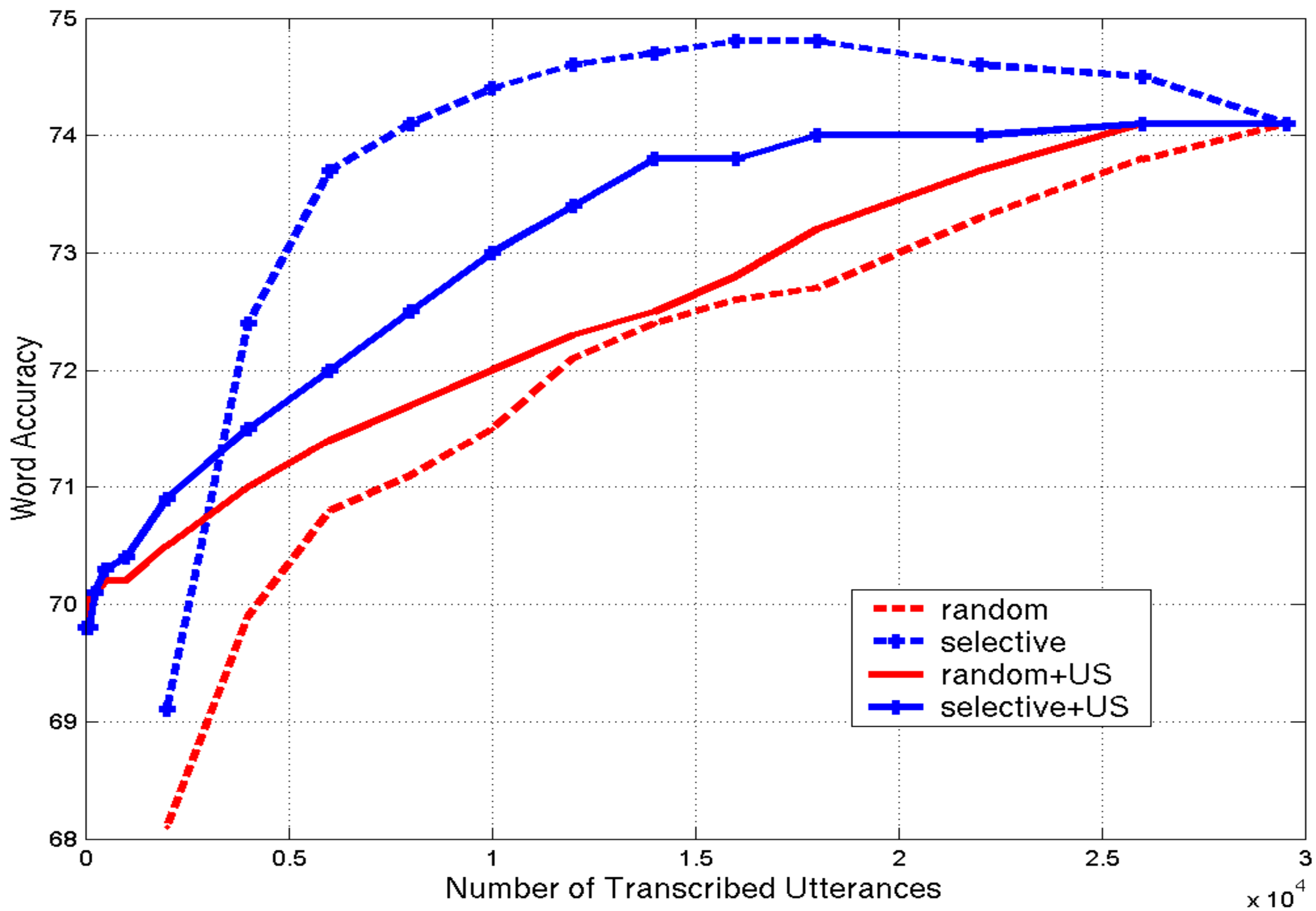
# Active and Unsupervised Learning Expt

- Initial Transcribed Data: Data collected from web, and Switchboard corpus.
- Additional Training Data: ~30K utterances from the HMIHY?SM
- Test Data: 5,171 utterances

# Active and Unsupervised Learning Expt

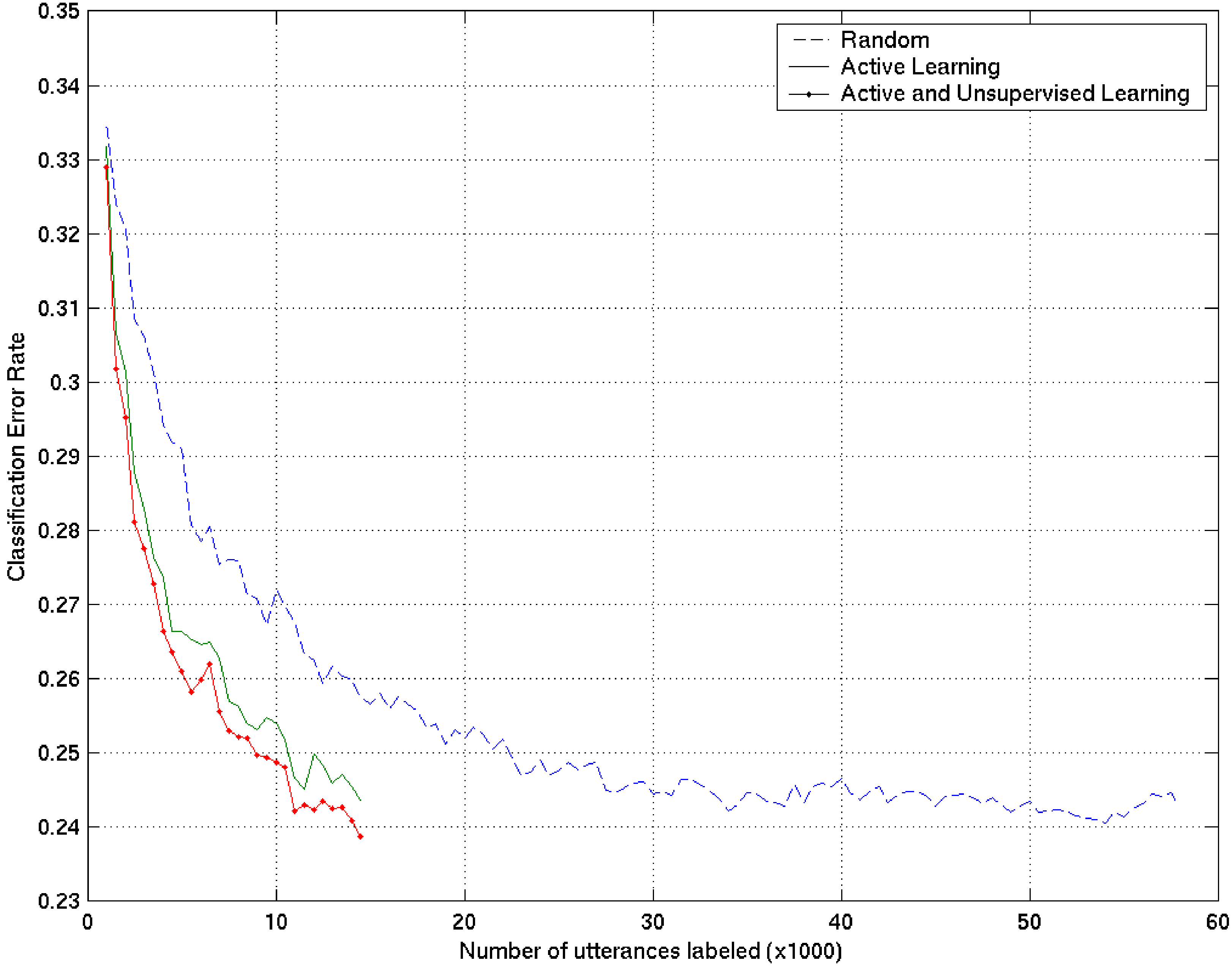


# Active and Unsupervised Learning Expt



# Call Classification

- *Tur, Hakkani-Tür, and Schapire, to appear.*
- 56 call types in total
- Dynamic Pool (1/4 of the candidate utterances selected at each iteration)
- Classifier: Boosting
- Combined Certainty-Based Active Learning with Unsupervised Learning



Random
Active Learning
Active and Unsupervised Learning
Classification Error Rate
Number of utterances labeled (x1000)
0.35
0.34
0.33
0.32
0.31
0.3
0.29
0.28
0.27
0.26
0.25
0.24
0.23
0
10
20
30
40
50
60

# Text Categorization

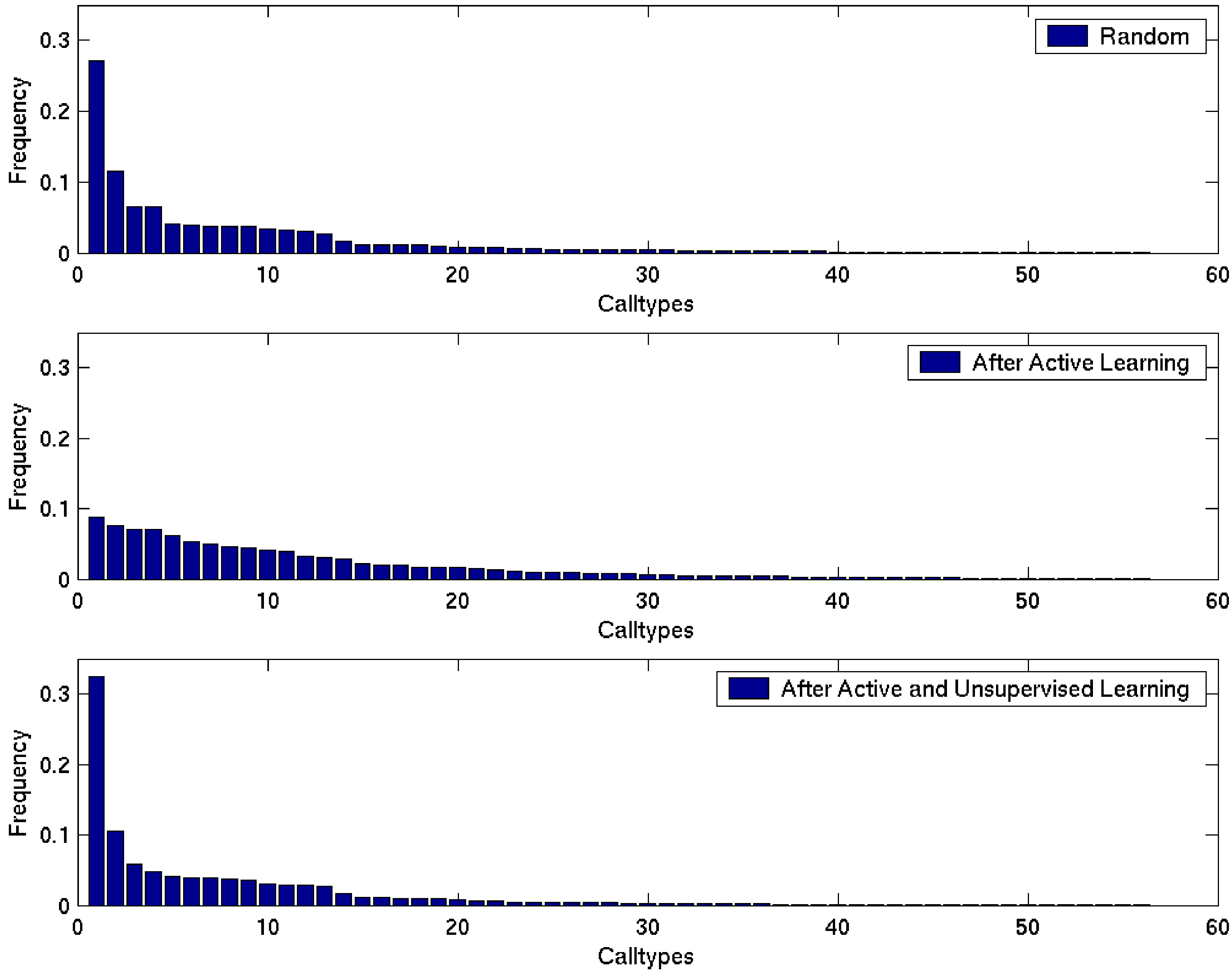
- *Muslea, Minton, and Knoblock, ICML'02*
- *Co-EMT algorithm:*

  Repeat N times
  - Run like *Co-EM* to get multiple learners
  - Run like *Committee-Based Active Learning* to decide on next data to label
- Outperformed both methods applied individually

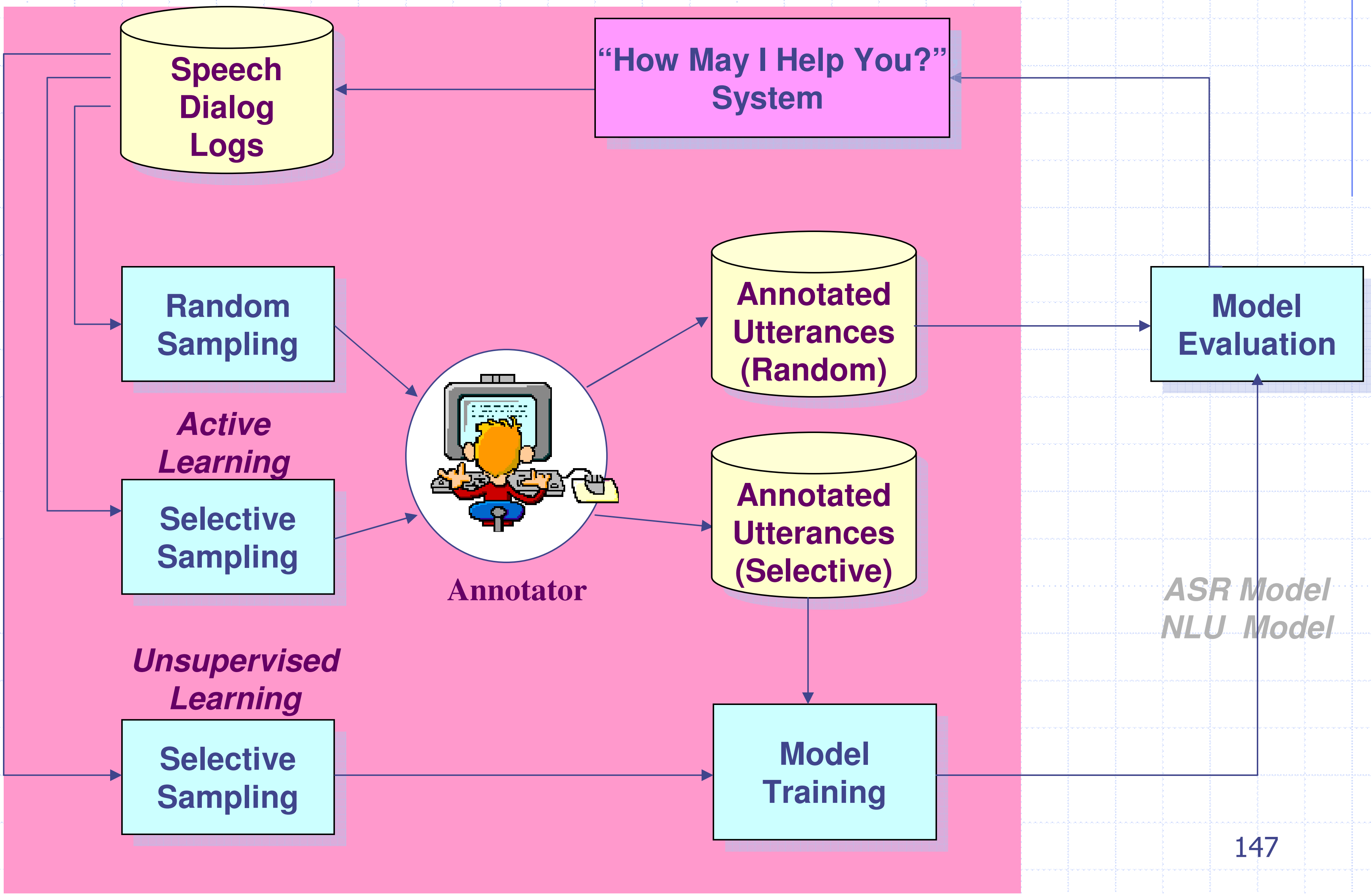
# Unbalanced Data Problem

- Unsupervised Learning changes the priors, too.
- Two issues may cancel each other, because:
  - Active Learning shaves more frequent classes
  - Unsupervised Learning do not favor infrequent classes
- Combining active and unsupervised learning may be a solution to both problems.

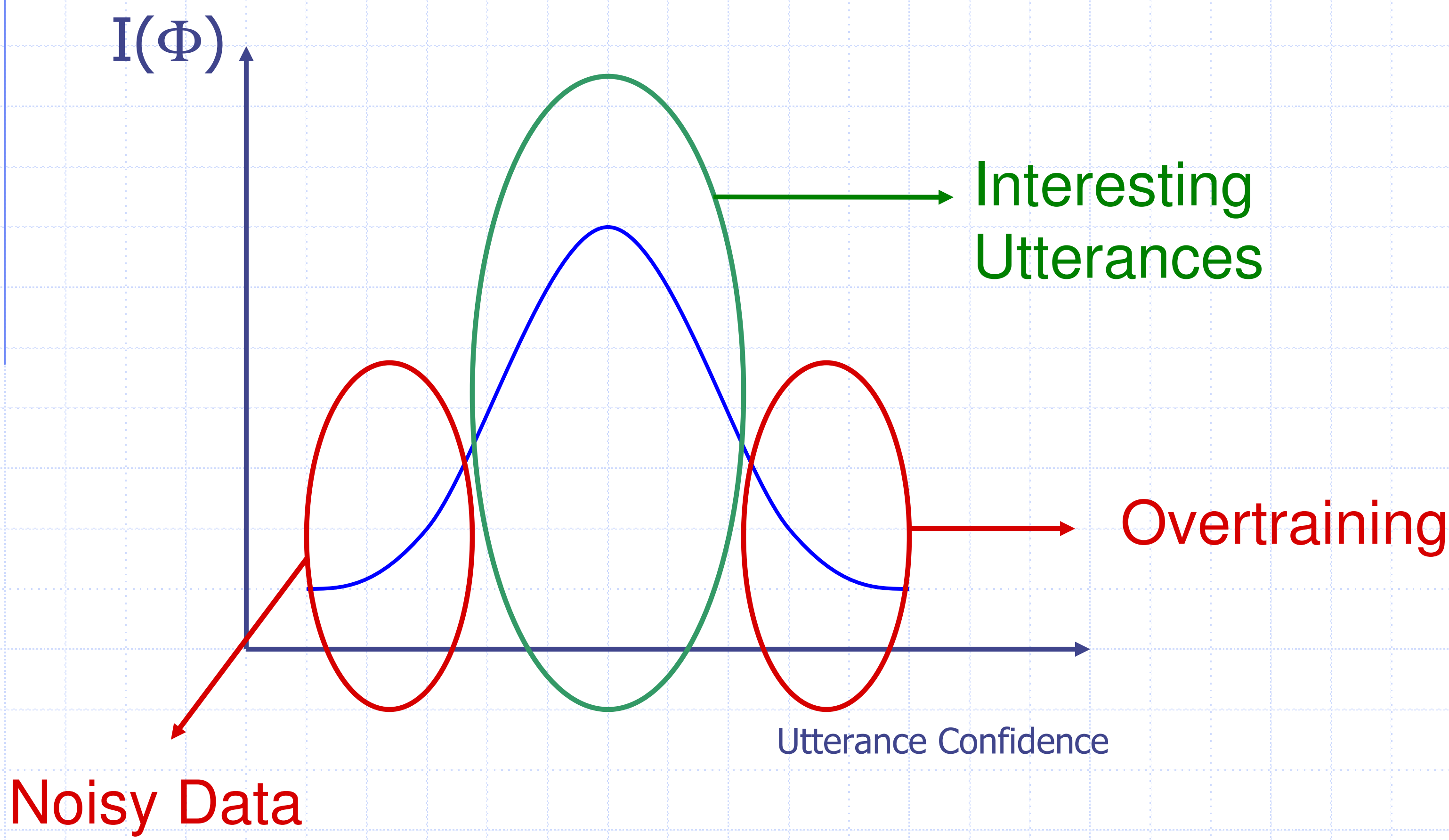
# UNBALANCED DATA PROBLEM



# Adaptive Learning in Practice



# Selective Sampling of Untranscribed Data



# Summary

- Adaptive Learning for Speech and Language Processing
  - Active Learning
    - Minimize human supervision by automatically selecting samples to be labeled
    - Optimize data for performance
  - Unsupervised Learning
    - Minimize human supervision by automatically labeling some of the data
    - Improve performance for free (finding unlabeled data is generally not an issue)
  - Combining active and unsupervised learning into a single and dynamic framework

# Open Research Issues

- Selective Sampling and Ranking algorithms
- Predict model error based on selected samples
- AL as optimization problem

# Bibliography

**Automatic Speech Recognition and Speech Understanding**
- L. Rabiner and B.-H. Juang. ***Fundamentals of Speech Recognition***. Prentice Hall. 1993.
- R. De Mori. ***Spoken Dialogues with Computers***. Academic Press. 1998.
- F. Jelinek. ***Statistical Methods for Speech Recognition.*** *MIT Press. 1997.*
- T. Mitchell. ***Machine Learning.*** *McGraw-Hill 1997*.
- Duda and P. Hart ***Pattern Classification and Scene Analysis.*** John Wiley & Sons. 1973

**Machine Learning**
- T. Hastie, R. Tibshirani and J. H. Friedman. ***The Elements of Statistical Learning: Data Mining****, Inference and Prediction.* Springer Verlag. 2001.
- Robert E. Schapire. ***The boosting approach to machine learning: An overview.*** Proceedings of the MSRI Workshop on Nonlinear Estimation and Classification, 2002.
- N. Christianini, J. Shawe-Taylor. ***An Introduction to Support Vector Machines and other kernel-based learning methods.*** Cambridge University Press. 2000.

**Active Learning (General)**
- D.D. Lewis and J. Catlett. ***Heterogeneous Uncertainty Sampling for Supervised Learning***. Proc. of the 11th International Conference on Machine Learning. 1994.
- D. Cohn and L. Atlas and R. Ladner. ***Improving Generalization with Active Learning***. Machine Learning. 1994.
- I. Dagan and S.P. Engelson. ***Committee-Based Sampling for Training Probabilistic Classifiers.*** Proc. of the 12th International Conference on Machine Learning. 1995.

# Bibliography

**Active Learning with Application to Automatic Speech Recognition**

- Dilek Hakkani-Tür, Giuseppe Riccardi, Allen Gorin. ***Active Learning for Automatic Speech Recognition.*** In the Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP 2002). 2002.
- T.M. Kamm and G.G.L. Meyer. ***Selective Sampling of Training Data for Speech Recognition.*** Proceedings of Human Language Technology Conference. 2002.

**Active Learning with Application to Natural Language Understanding**

- Gokhan Tur, Robert E. Schapire, and Dilek Hakkani-Tür. ***Active Learning for Spoken Language Understanding.*** Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP'03). 2003.
- R. Liere and P. Tadepalli. ***The Use of Active Learning in Text Categorization.*** Working Notes of the AAAI, Spring Symposium on Machine Learning in Information Access. 1996.

**Unsupervised Learning with Application to Automatic Speech Recognition**

- R. Gretter and G. Riccardi. ***On-line Learning of Language Models with Word Error Probability Distributions.*** Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing. 2001.
- T. Kemp and A. Waibel. ***Learning to Recognize Speech by Watching Television.*** IEEE Intelligent Systems. 1999.
- A. Stolcke. ***Error Modeling and Unsupervised Language Modeling.*** Proceedings of the 2001 NIST Large Vocabulary Conversational Speech Recognition Workshop. 2001.
- G. Riccardi and D. Hakkani-Tür. ***Active and Unsupervised Learning for Automatic Speech Recognition.*** Submitted.

# Bibliography

**Unsupervised Learning with Application to Natural Language Understanding**

- K. Nigam, A. McCallum, S. Thrun and T. Mitchell. ***Text Classification from Labeled and Unlabeled Documents using EM.*** Machine Learning. Volume 39. Pages: 103-134. 2000.
- R. Ghani. ***Combining Labeled and Unlabeled Data for Multiclass Text Categorization.*** Proceedings of the 19th International Conference on Machine Learning (ICML-02). 2002.
- A. Blum and T. Mitchell. C***ombining Labeled and Unlabeled Data with Co-Training.*** Proceedings of the Workshop on Computational Learning Theory (COLT). 1998.
- G. Tur and D. Hakkani-Tür. ***Exploiting Unlabeled Utterances for Spoken Language Understanding.*** Submitted.