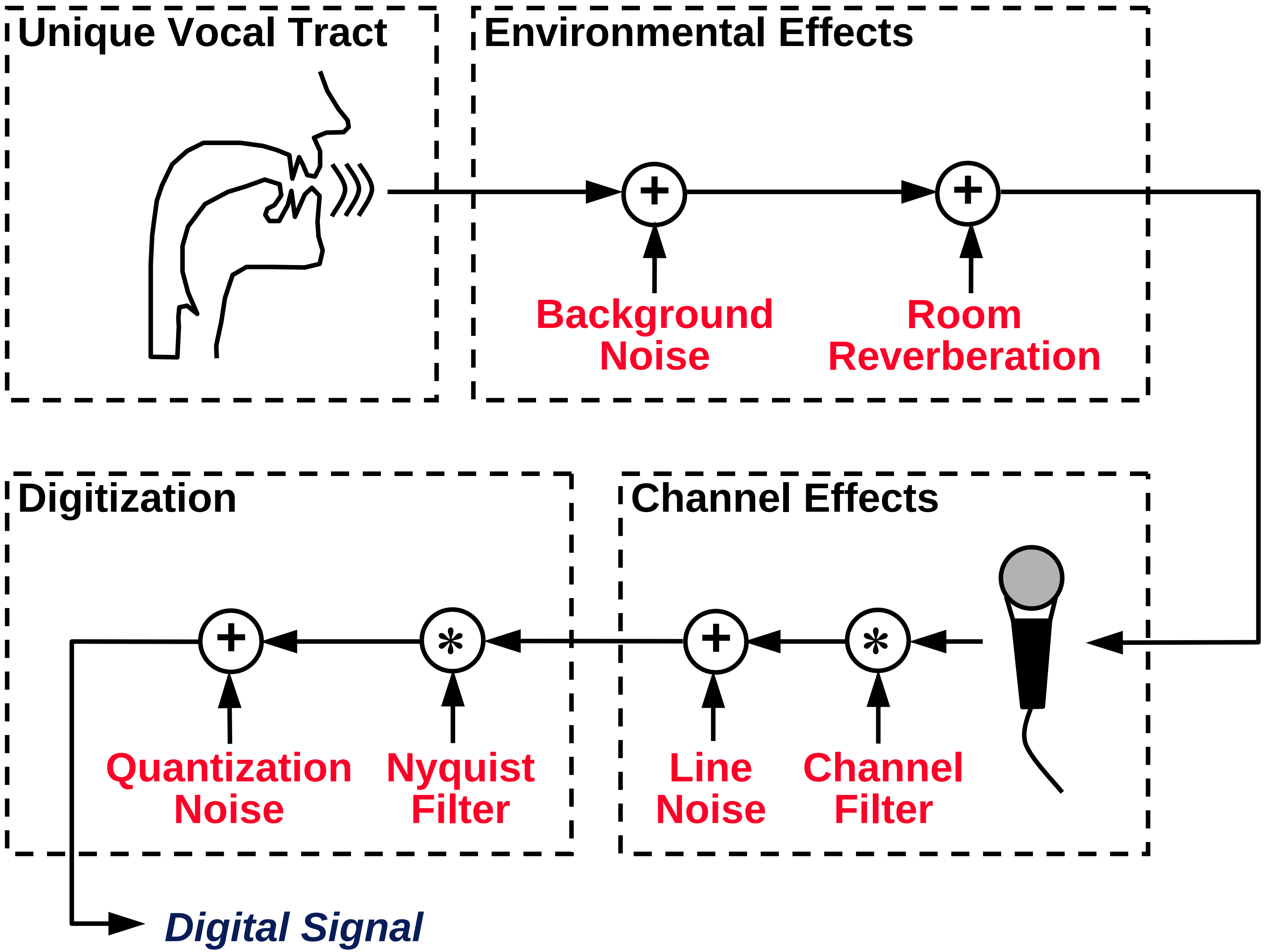
# Noise Robustness and Confidence Scoring

Lecturer: T. J. Hazen

- Handling variability in acoustic conditions
  - Channel compensation
  - Background noise compensation
  - Foreground noises and non-speech artifacts
- Computing and applying confidence scores
  - Recognition confidence scoring
  - Language understanding issues
  - Dialogue modeling issues

# Typical Digital Speech Recording



# Motivation

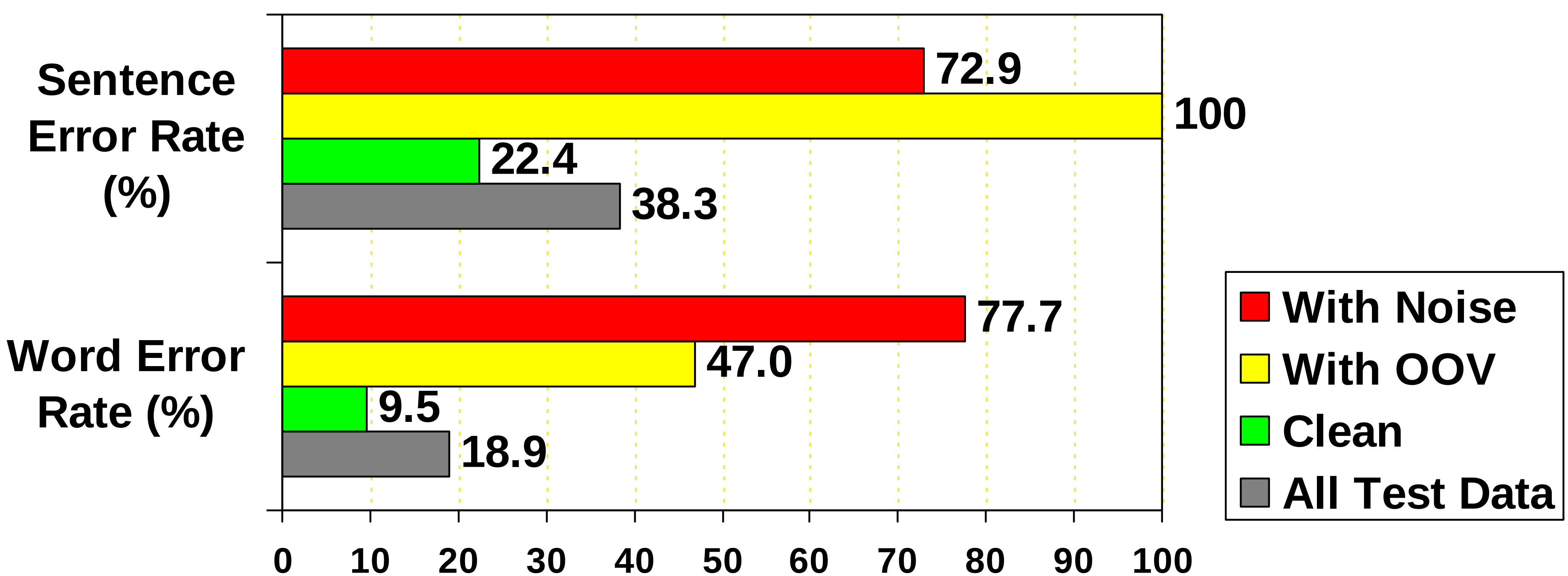
- Recognizers make errors
- Some reasons for errors:
  - Presence of previously unseen words or events
  - Difficult acoustic conditions or background noises
  - Presence of highly confusable words
  - Insufficient amount of training data
  - Mismatch between training and testing data
  - Models too rigid to handle variability
- Methods to handling error-full data
  - Adjust or adapt to current conditions
  - Identify when errors occur and perform action to recover

# Noises and Non-Speech Artifacts

- **Non-speech artifacts can be extremely varied**
  - **Background noises (music, dog bark, door slam, etc.)**
  - **Microphone and channel noises (clicks, beeps, static, etc.)**
  - **Non-lexical speaker noises (cough, laugh, lip smack, etc.)**
- **Noises can be simultaneous with speech**

# Recognition Experiments

- **Experiments w/ baseline JUPITER recognizer**
  - **Clean ➜ No OOV words and no non-speech artifacts**
  - **With Noise ➜ Contains at least one non-speech artifact**
  - **With OOV ➜ Contains at least one OOV word**



# Difficult Channel and Noise Conditions

- **Variable system functions**
  - **From different channels (e.g., land line, cellular, etc.)**
  - **Different microphones**
- **Constant background noise**
  - **Channel static**
  - **Car engine noise**
  - **Air conditioning hiss**
- **Intermittent foreground or background noises**
  - **Cough**
  - **Laugh**
  - **Door slam**
  - **Handset taps or clicks**
  - **Phone ringing**
  - **Dog barking**

# Cepstral Mean Normalization

- **The channel of a speech recording can be modeled as a linear-time invariant filter:**

$$\underset{\substack{\uparrow \\ \boxed{\text{recorded speech}}}}{y[n]} = \underset{\substack{\uparrow \\ \boxed{\text{original speech}}}}{s[n]} * \underset{\substack{\uparrow \\ \boxed{\text{channel filter}}}}{f[n]}$$

- **In the frequency domain this becomes:**

$$Y(\omega) = S(\omega)F(\omega)$$

- **In the log-frequency domain this becomes:**

$$\log Y(\omega) = \log S(\omega) + \log F(\omega)$$

- **In the cepstral domain this becomes:**

$$c[n] = \hat{s}[n] + \hat{f}[n]$$

# Cepstral Mean Normalization (cont)

- **During recognition, speech is processed in frames**
- **Let c[n,m] be the nth cepstral coefficient of the mth frame:**

$$c[n,m] = \hat{s}[n,m] + \hat{f}[n,m]$$

- **Because the channel filter is linear time invariant:**

$$\hat{f}[n,m] = \hat{f}[n] \quad \Rightarrow \quad c[n,m] = \hat{s}[n,m] + \hat{f}[n]$$

- **Goal: Remove the effect of the filter!**
- **Start by averaging cepstrum over all frames:**

$$\overline{c}[n] = \frac{1}{M}\sum_{m=1}^{M} c[n,m] = \hat{f}[n] + \frac{1}{M}\sum_{m=1}^{M} \hat{s}[n,m]$$

# Cepstral Mean Normalization (cont)

- **Cepstral mean normalization is:**

$$c'[n,m] = c[n,m] - \overline{c}[n]$$

$$= \left(\hat{s}[n,m] + \boxed{\hat{f}[n]\right) - \left(\hat{f}[n]} + \frac{1}{M}\sum_{m=1}^{M}\hat{s}[n,m]\right)$$

**Filter properties are removed**

$$= \hat{s}[n,m] - \boxed{\frac{1}{M}\sum_{m=1}^{M}\hat{s}[n,m]}$$

**Average cepstrum of speech is also removed**

- **Useful when filter variation is larger than speaker variation**
  - **Reference: Furui, 1981**

# Handling Background Noise

- **Multi-style training**
  - **Train with data from a variety of noisy environments**
  - **Problem: Poor estimates for new or unexpected environments**
  - **Reference: Lippmann, *et al*, 1987**
- **Spectral-subtraction**
  - **Estimate static spectral components during silence**
  - **Subtract static spectral components from dynamic spectra**
  - **Problem: Poor estimates of speech in regions with low signal-to-noise ratio**
  - **Reference: Boll, 1979**
- **Sub-band recognition**
  - **Run parallel "sub-band" recognizers**
  - **Sub-band recognizers operate on different spectral bands**
  - **Weight sub-bands based on their signal-to-noise ratio**
  - **Problem: Using multiple recognizers is computationally expensive**
  - **Reference: Bourlard and Dupont, 1996**

# Parallel Model Combination

- **Parallel Model Combination (PMC) for background noise compensation**
  - **Train speech acoustic models on clean speech**
  - **Estimate noise model for current conditions**
  - **Combine clean speech models with estimated noise model**
- **Method assumes mean spectrum of signal can be reverse estimated from mean vector of model**
  - **Clean speech model for phonetic unit u:**

$$P(\vec{s} \mid u) \equiv N(\vec{\mu}_u, \Sigma_u) \Longrightarrow S(\omega) = F^{-1}(\vec{\mu}_u)$$

  - **Noise model estimated from non-speech region of current conditions:**

$$P(\vec{n}) \equiv N(\vec{\mu}_n, \Sigma_n) \Longrightarrow N(\omega) = F^{-1}(\vec{\mu}_n)$$

# Parallel Model Combination

- **Given estimates of the mean spectral values of clean speech and noise, do combination:**

$$\vec{\mu}'_u = F\left(S(\omega) + N(\omega)\right) = F\left(F^{-1}(\vec{\mu}_u) + F^{-1}(\vec{\mu}_n)\right)$$

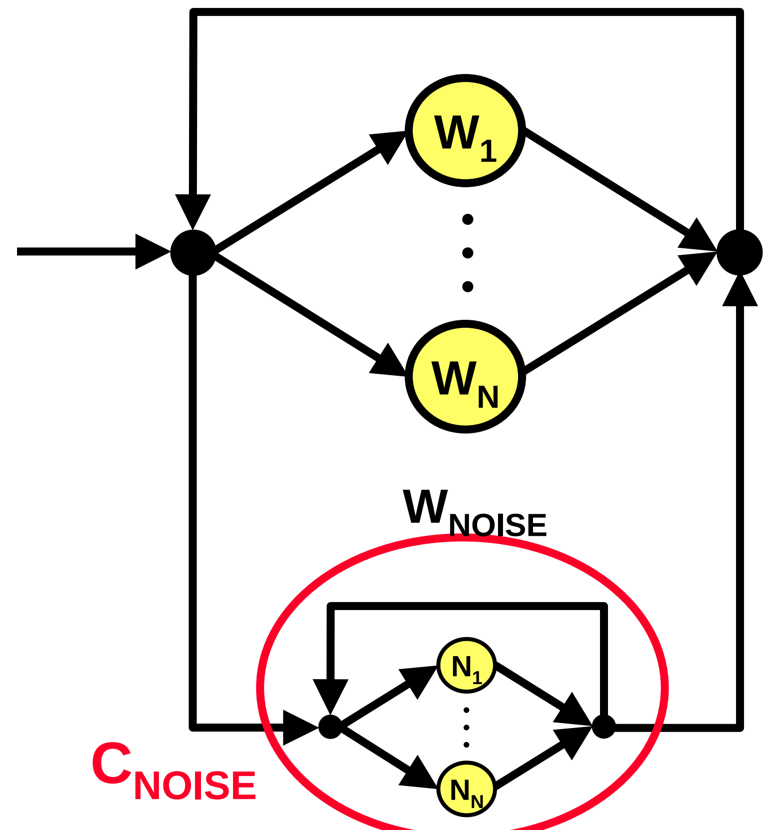
$$P_{PMC}(\vec{a} \mid u) \equiv N(\vec{\mu}'_u, \Sigma_u)$$

- **Issues:**
  - **Must be able to reverse estimate spectrum from model mean**
  - **Must have a reliable estimate of current noise conditions**
- **Reference: Gales, 1996**

# Handling Foreground Noises

- **Build explicit models for different noises and non-speech artifacts**
  - **Reference: Ward, 1989**

- **One possible approach:**
  - **Build acoustic model network for each noise model**
  - **Noise network contains multiple states to model dynamic noises**
  - **Add noise networks to word network as new words**
  - **Control noise detection rate with cost, $C_{NOISE}$**



# Non-Speech Modeling Experiment

- **Added 5 non-speech models to JUPITER**
  - <cough>, <laugh>, <noise>, <background>, <hangup>
  - Reference: Hazen, Hetherington and Park, 2001
- **Word error rate results:**

| Test Set Data | Baseline | + Noise Models |
|---|---|---|
| All Data | 18.9% | 17.1% |
| Data w/ Noise | 64.0% | 45.1% |
| IV Data w/ Noise | 46.4% | 28.2% |
| IV Data w/ No Noise | 9.4% | 9.6% |

**IV = In-vocabulary data only**

# Confidence Scoring Overview

- **Question: How do we assess if a recognizer's hypothesis is correct or not?**
- **Goal: Generate confidence scores which estimate the likelihood that a hypothesis is correct**
- **Scores can be computed at multiple levels:**
  - **Phonetic scores**
  - **Word scores**
  - **Utterance scores**
- **One approach:**
  - **Find features correlated with correctness**
  - **Construct feature vector from good features**
  - **Build correct/incorrect classifier for feature vector**

# Acoustic Likelihood Scores

- **An acoustic likelihood score is computed as:**

$$p(\vec{x} \mid u)$$

- **Acoustic likelihood scores are good for comparing different hypotheses**
  - **Score are relative density likelihoods, not probabilities**
- **Likelihood scores do not provide good estimate of correctness or reliability**

# Normalized Acoustic Scores

- The *a posteriori* probability expression is:

$$p(u \mid \vec{x}) = \boxed{\frac{p(\vec{x} \mid u)}{p(\vec{x})}} p(u)$$

**normalized acoustic likelihood score**

- In probabilistic framework $p(\vec{x})$ is usually ignored
- Recognition is unaffected by normalization
  - normalization model is independent of phone identity
  - normalized scores can be viewed as confidence scores

# Normalized Acoustic Scores

- **Theoretically normalization model is:**

$$p(\vec{x}) = \sum_{\forall u} p(\vec{x} \mid u)\, p(u)$$

- **In practice normalization is performed with an approximate model of $p(\vec{x})$**
- **Approximation of $p(\vec{x})$ using bottom-up clustering:**
  - **Similar Gaussian components merged**
  - **Merged model is ML approximation of mixture components to be merged**
  - **Merging continues until desired size is reached**
  - **Normalization model typically has between 50 and 100 mixture components in SLS recognizers**

# Word Confidence Features

- Want to extract information from recognition computation which is correlated with correctness
- Possible word level confidence features extracted from acoustic scores:
  - Mean normalized acoustic score over word
  - Minimum normalized acoustic score over word
  - Mean normalization model score
- Other sources of information:
  - N-best purity scores
  - Language model scores
  - Number of competing hypotheses
  - Relative score differences between hypotheses
- Reference: Chase, 1997

# The *N*-best Purity Measure

- ***N*-best purity is the fraction of *N*-best hypotheses in which a word hypothesis appears**

| | | | | | | |
|---|---|---|---|---|---|---|
| **(1)** | ***what*** | ***is*** | ***the*** | ***weather*** | ***in*** | ***new york*** |
| | 1.0 | 0.8 | 0.6 | 1.0 | 1.0 | 0.6 |
| **(2)** | ***what*** | ***is*** | ***the*** | ***weather*** | ***in*** | ***newark*** |
| | 1.0 | 0.8 | 0.6 | 1.0 | 1.0 | 0.4 |
| **(3)** | ***what*** | ***is*** | ***<uh>*** | ***weather*** | ***in*** | ***new york*** |
| | 1.0 | 0.8 | 0.4 | 1.0 | 1.0 | 0.6 |
| **(4)** | ***what*** | ***is*** | ***<uh>*** | ***weather*** | ***in*** | ***newark*** |
| | 1.0 | 0.8 | 0.4 | 1.0 | 1.0 | 0.4 |
| **(5)** | ***what*** | ***was*** | ***the*** | ***weather*** | ***in*** | ***new york*** |
| | 1.0 | 0.2 | 0.6 | 1.0 | 1.0 | 0.6 |

**newark is in 2 of 5 hypotheses ॐpurity = 2/5 = 0.4**

# Confidence Classification

- **Given a confidence feature vector we want to classify the vector as *correct* or *incorrect***
- **This is a standard two class classification problem**
- **Possible approaches:**
  - **Linear discriminant projection (Pao, *et al*, 1998)**
  - **Neural network classifier (Wendemuth, *et al*, 1999)**
  - **Mixture Gaussian classifier (Kamppari & Hazen, 2000)**
  - **Support vector machines (Ma, *et al*, 2001)**

# Linear Discriminant Classifier

- **Discriminative linear projection applied to confidence feature vector:**

$$\mathbf{r} = \vec{\mathbf{p}}^{\,t}\vec{\mathbf{f}}$$

**raw score**

**feature vector**

**projection vector**

- **Projection vector:**
  - **Trained on independent development set**
  - **Minimum Classification Error (MCE) training**
  - **MCE performs gradient descent training on error rate**

# Probabilistic Confidence Classifier

- **MAP-based classifier trained for raw score:**

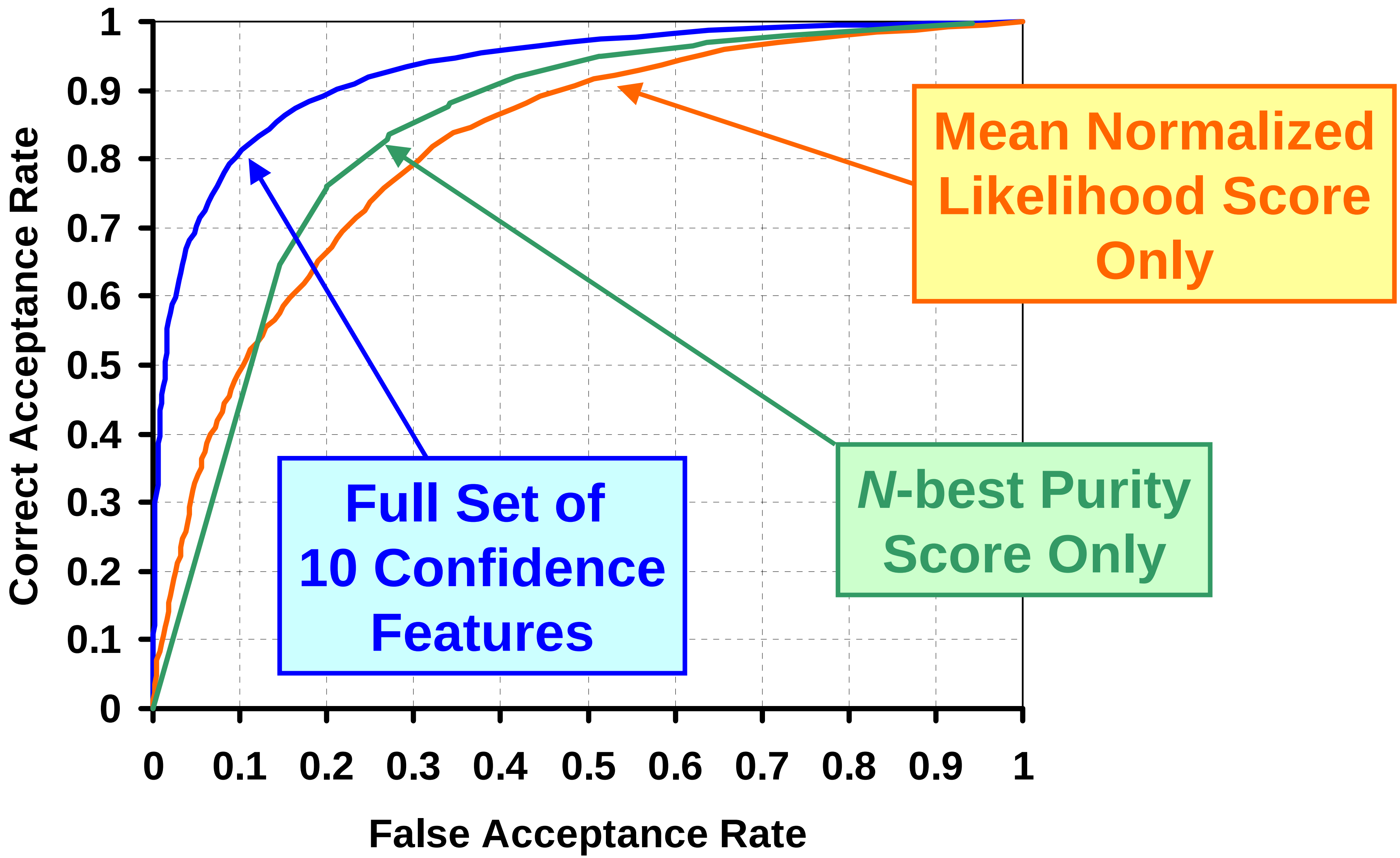
$$c = \log\left(\frac{p(r \mid \text{correct})P(\text{correct})}{p(r \mid \text{incorrect})P(\text{incorrect})}\right) - t$$

- **Probabilistic model:**
  - **Trained on independent set of development data**
  - **Gaussian models can be used for likelihood densities**
  - **Priors based on recognizer hypothesis error rate**
- **Threshold can be varied to adjust balance of *false acceptances* vs. *false rejections***

# Word Confidence Experiment

- **Want to reject hypothesized words for which recognizer has low confidence**
- **Train confidence model on independent development data**
- **Test on independent test set of JUPITER data**
- **Evaluate using ROC curve**
  - **Examines correct acceptances vs. false acceptances**
  - **Want to reject incorrectly hypothesized words and accept correctly hypothesized words**
  - **Results shown for two individual feature and for full feature vector with 10 features**
- **Reference: Hazen, *et al*, 2002**

# Word Confidence Results



# Using Confidence Scores

- **To be useful, confidence scores must be integrated with language understanding and dialogue modeling**
- **Confidence scores are often quantized into two or three decision regions:**
  - **Accept or reject (two regions)**
  - **Accept, reject, or uncertain (three regions)**
- **Language understanding component can be adapted to handle rejected words**
- **Dialogue management component can perform different actions based on confidence score**
  - **Perform normal action when everything is accepted**
  - **Ask for confirmation when uncertain**
  - **Ask user to repeat or rephrase when rejected**
- **Reference: Hazen, *et al*, 2002**

# *N*-best List Modifications

***What is the forecast for Paramus Park, New Jersey?***

## Standard *N*-best list with confidence scores:

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| what_is | 6.13 | the | 5.48 | forecast | 6.88 | for | 5.43 | paris | -0.03 | park | 4.41 | new_jersey | 4.35 |
| what_is | 6.13 | the | 5.48 | forecast | 6.88 | for | 4.47 | hyannis | -0.61 | park | 4.41 | new_jersey | 4.35 |
| what_is | 6.13 | the | 5.48 | forecast | 6.88 | for | 5.12 | venice | -0.89 | park | 4.41 | new_jersey | 4.35 |
| what_is | 6.13 | the | 5.48 | forecast | 6.88 | for | 4.28 | france | -1.12 | park | 4.41 | new_jersey | 4.35 |

## *N*-best list with *hard rejection* of low scoring words:

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| what_is | 6.13 | the | 5.48 | forecast | 6.88 | for | 5.43 | *reject* | 0.00 | park | 4.41 | new_jersey | 4.35 |
| what_is | 6.13 | the | 5.48 | forecast | 6.88 | for | 4.47 | *reject* | 0.00 | park | 4.41 | new_jersey | 4.35 |
| what_is | 6.13 | the | 5.48 | forecast | 6.88 | for | 5.12 | *reject* | 0.00 | park | 4.41 | new_jersey | 4.35 |
| what_is | 6.13 | the | 5.48 | forecast | 6.88 | for | 4.28 | *reject* | 0.00 | park | 4.41 | new_jersey | 4.35 |

# *N*-best List Modifications (cont.)

## *N*-best list with *optional rejection:*

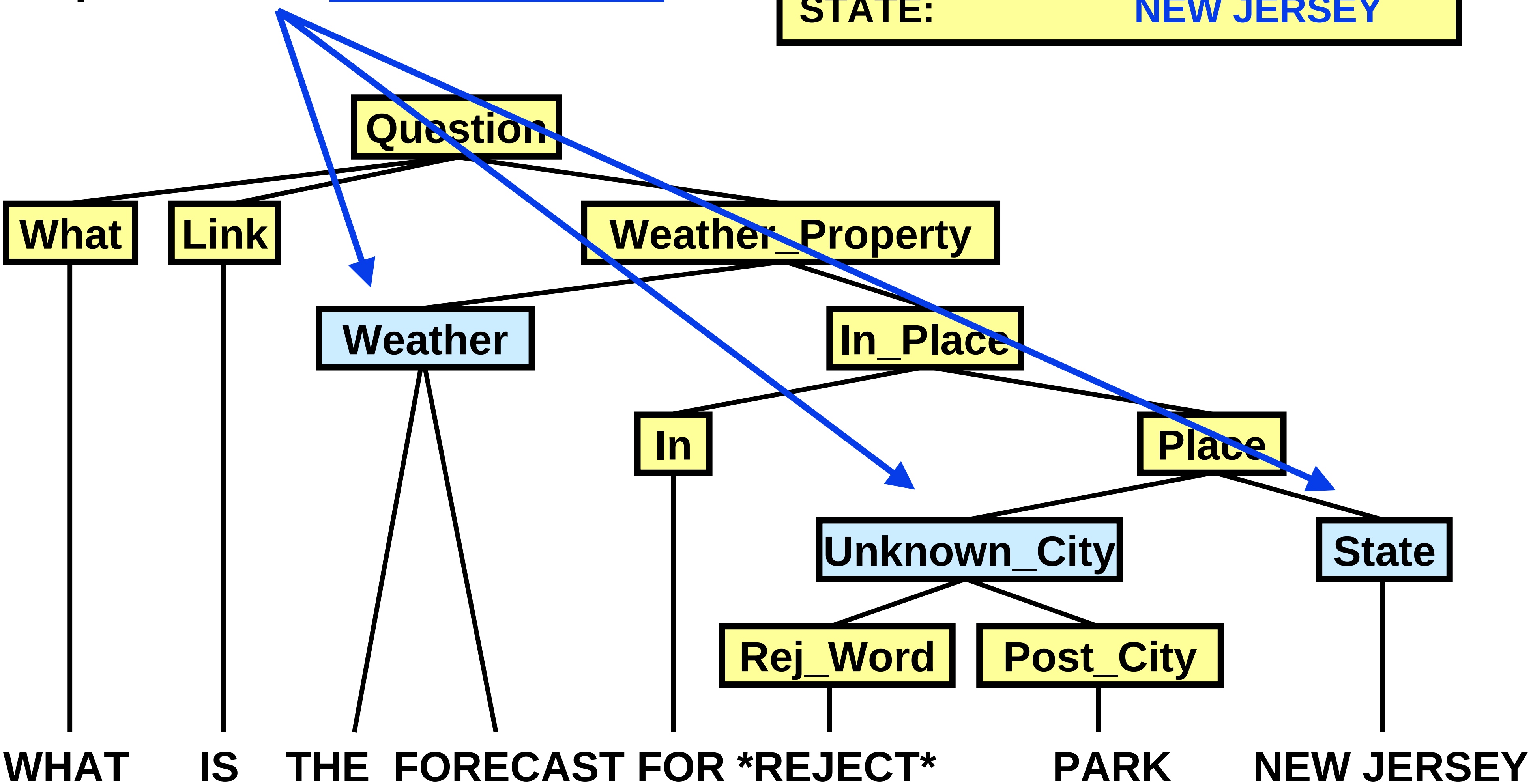
| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| what_is | 6.13 | the | 5.48 | forecast | 6.88 | for | 5.43 | paris | -0.03 | park | 4.41 | new_jersey | 4.35 |
| what_is | 6.13 | the | 5.48 | forecast | 6.88 | for | 5.43 | *reject* | 0.00 | park | 4.41 | new_jersey | 4.35 |
| what_is | 6.13 | the | 5.48 | forecast | 6.88 | for | 4.47 | hyannis | -0.61 | park | 4.41 | new_jersey | 4.35 |
| what_is | 6.13 | the | 5.48 | forecast | 6.88 | for | 4.47 | *reject* | 0.00 | park | 4.41 | new_jersey | 4.35 |
| what_is | 6.13 | the | 5.48 | forecast | 6.88 | for | 5.12 | venice | -0.89 | park | 4.41 | new_jersey | 4.35 |
| what_is | 6.13 | the | 5.48 | forecast | 6.88 | for | 5.12 | *reject* | 0.00 | park | 4.41 | new_jersey | 4.35 |
| what_is | 6.13 | the | 5.48 | forecast | 6.88 | for | 4.28 | france | -1.12 | park | 4.41 | new_jersey | 4.35 |
| what_is | 6.13 | the | 5.48 | forecast | 6.88 | for | 4.28 | *reject* | 0.00 | park | 4.41 | new_jersey | 4.35 |

**Words with poor confidence scores compete with rejected words during natural language understanding search**

# Example Understanding Parse Tree

**Semantic concepts extracted expressed as key-value pairs**

| | |
|---|---|
| WEATHER: | FORECAST |
| UNKNOWN_CITY: | *REJECT* PARK |
| STATE: | NEW JERSEY |



# Understanding Evaluation

- **Confidence scoring rejection evaluated using understanding *concept error rate* (CER)**
- **CER computed from substitutions, insertions, and deletions of concept key-value pairs**

| **Reference Key-Values:** | |
|---|---|
| **WEATHER:** | **forecast** |
| **COUNTRY:** | **bosnia** |
| **DATE:** | **sunday** |

| **Hypothesis Key-Values:** | |
|---|---|
| **WEATHER:** | **forecast** |
| **DATE:** | **monday** |
| **CITY:** | **boston** |

**Deletion Error**

**Substitution Error**

**Insertion Error**

# Understanding Results

- Experimental understanding results on JUPITER
- Original system did not use confidence scores during hypothesis selection in understanding component

| Experimental Conditions | Concept Error Rates (CER) (%) | | | |
|---|---|---|---|---|
| | Sub | Ins | Del | Total |
| Original System | 2.2 | 19.9 | 6.3 | 28.5 |
| New System/No Rej. | 2.1 | 18.1 | 6.1 | 26.2 |
| + optional rejection | 1.3 | 8.9 | 8.5 | 18.7 |
| + hard rejection | 1.0 | 7.0 | 10.6 | 18.6 |

35% reduction in CER

Sub = Substitution

Ins = Insertion

Del = Deletion

# Dialogue Modeling Examples

**Aiding the user with additional information:**

- **USER:** ***what is the forecast for *reject* park new jersey***

- **KEY_VALUES:** **WEATHER: forecast**
  **UNKNOWN_CITY: *reject* park**
  **STATE: new jersey**

- **RESPONSE:** ***I may not know the city you are asking about. I know the following places in New Jersey: Atlantic City, Camden, Newark, Piscataway, Princeton, and Trenton. Please select one.***

# More Dialogue Modeling Examples

**Guiding the user towards a query that can be handled:**

- **USER:** ***what is the forecast for *reject* *reject****

- **KEY_VALUES:** **WEATHER: forecast**
  **UNKNOWN_CITY: *reject* *reject***

- **RESPONSE:** ***I may not know the city you are asking about. Try asking for the state or country. For example: "What cities do you know about in Massachusetts?"***

# More Dialogue Modeling Examples

**Asking for confirmation:**

- **<u>USER:</u>** ***what is the forecast for *boston****

- **<u>KEY_VALUES:</u>** **WEATHER: forecast**
  **CONFIRM_CITY: boston**

- **<u>RESPONSE:</u>** ***Did you say Boston?***

# References


- **S. Boll, "Suppression of acoustic noise in speech using spectral subtraction,"** ***IEEE Trans. on Acoustic, Speech, and Signal Processing*****, April 1979.**
- **H. Bourlard and S. Dupont, "A new ASR approach based on independent processing and recombination of partial frequency bands," ICSLP, 1996.**
- **L. Chase, "Word and acoustic confidence annotation for large vocabulary speech recognition," Eurospeech, 1997.**
- **S. Furui, "Cepstral analysis techniques for automatic speaker verification,"** ***IEEE Trans. on Acoustic, Speech, and Signal Processing*****, April 1981.**
- **M. Gales and S. Young, "Robust continuous speech recognition using parallel model combination,"** ***IEEE Trans. on Acoustic, Speech, and Signal Processing*****, September 1996.**
- **T. Hazen, L. Hetherington, and A. Park, "FST-based recognition techniques for multi-lingual and multi-domain spontaneous speech," Eurospeech, 2001.**
- **T. Hazen, J. Polifroni and S. Seneff, "Recognition confidence scoring for use in speech understanding systems,"** ***Computer Speech and Language*****, January, 2002.**

# References


- C. Pao, P. Schmid, and J. Glass, “Confidence scoring for speech understanding,” ICSLP, 1998.
- S. Kamppari and T. Hazen, “Word and phone level acoustic confidence scoring,” ICASSP, 2000.
- R. Lippman, E. Martin, and D. Paul, “Multi-style training for robust isolated-word speech recognition,” ICASSP, 1987.
- C. Ma, M. Randolph, and J. Drish, “A support vector machine-based rejection technique for speech recognition,” ICASSP, 2001.
- W. Ward, “Modelling non-verbal sounds for speech recognition,” DARPA Speech and Natural Language Workshop, October, 1989.
- A. Wendemuth, G. Rose, and J. Dolfing, “Advances in confidence measures for large vocabulary,” ICASSP, 1999.