HTK Version 3.4 Features (cont)

Mark Gales, Andrew Liu & Phil Woodland

19th April 2007

HTK3 Development Team
Cambridge University Engineering Department

HTK users meeting ICASSP’07
**HTK Large Vocabulary Decoder - HDecode**

- **Basic Features:**
  - bi-gram or tri-gram full decoding
  - lattice generation
  - lattice rescoring and alignment

- **Supporting many other HTK Features:**
  - fully integrated with adaptation schemes
  - STC and HLDA
  - lattice generation for discriminative training

- **Typical use in a multi-pass system**

- **Limitations and Future Development**
**HDecode: Basic Features (1)**

- Tree structure network based beam search cross-word trip-hone decoder.

- Effective pruning techniques to constrain search space:
  - main search beam
  - word end beam
  - maximum active model
  - lattice beam
  - LM back-off beam

- Efficient likelihood computation during decoding:
  - state and/or component output probability caching
  - language model probability caching

- Token sets merging and LM score look-ahead during propagation
HDecode: Basic Features (2)

HDecode performs search using a model level network expanded from a dictionary and a finite state grammar constructed from a word based bi-gram or tri-gram model, as in **full decoding**:

- 1-best transcription stored in HTK MLF format.

- word lattices may be generated in HTK SLF format with
  - detailed timing
  - word level scores (acoustic, LM and pron)
  - LM and pron prob scaling factors
  - other model specific information

- Higher order N-gram models applicable to resulting lattices (HLRescore).
HDecode: Basic Features (3)

or word lattices marked with LM scores, as in lattice rescoring.

- HDecode outputs “word lattices” containing duplicate word paths of
  - different pronunciation variants - “contrapoint”
  - silence related different phone contexts - “fugue”

- **determinization** of word lattices required prior to rescoring (HLRescore).

- 1-best hypothesis and lattices generated as in full decoding.

- model level alignment may also be generated in resulting lattices:
  - model alignment and duration marked on lattice arcs
  - important for discriminative training
HDecode: Supported new HTK Features

- A variety forms of linear transformations for adaptation:
  - MLLR transforms
  - CMLLR transforms
  - covariance transforms
  - hierarchy of linear transformations

- Covariance modeling and linear projection schemes:
  - STC
  - HLDA

- Lattice generation for discriminative training:
  - denominator word lattices generation
  - numerator and denominator lattices model alignment
**HDecode: Typical use in a multi-pass system**

- Upadapted tri-gram decoding plus 4-gram rescoring to generate initial hypotheses with tight pruning.

- Bi-gram or tri-gram adapted full decoding to generate word lattices with wide pruning.

- Lattice expansion and pruning using more complicated LMs (HLRescore).

- Lattice rescoring using re-adapted more complicated acoustic models and system combination.
HDecode: Limitations and Future Development

- Known limitations are:
  - only works for cross-word tri-phones;
  - sil and sp symbols reserved for silence models;
  - appended to all words in pronunciation dictionary;
  - lattices generated require determinization for rescoring;
  - only batch mode adaptation supported.

- Possible future work areas:
  - fast Gaussian likelihood computation?
  - more efficient token pruning?
  - incremental adaptation?
**HTK Discriminative Training Tools**

- **Basic Features:**
  - MMI
  - MPE and MWE
  - efficient lattice based implementation

- **Supporting many other HTK Features:**
  - fully integrated with adaptation schemes
  - discriminative MAP
  - lattice based adaptation
  - single pass re-train using new front-ends

- **Typical procedure of building discriminatively trained models**
Two types of discriminative training criteria supported:

- maximum mutual information (MMI)
  \[ F(\lambda) = \sum_r \log P(W^r|O^r, \lambda) \]

- minimum Bayes risk (MBR)
  \[ F(\lambda) = \sum_{r, \tilde{W}} P(\tilde{W}^r|O^r, \lambda) A(W, \tilde{W}) \]

with error cost function \( A(W, \tilde{W}) \) computed on

- phone model level - minimum phone error (MPE)
- word level - minimum word error (MWE)
HTK Discriminative Training Tools: Basic Procedure

Ref

LM

Audio

ML AM

HLRescore

Num Lat

HDecode

HLRescore

Den Lat

HMMIRest

MPE AM
HTK Discriminative Training Tools: I-smoothing

Flexible use of prior information for parameter smoothing:

- Common priors used in I-smoothing:
  - ML statistics
  - MMI statistics
  - Static model based priors
  - hierarchy of smoothing statistics back-off
  - important for MPE/MWE training to generalize well

- Applicable to a variety of systems:
  - useful in discriminative MAP training
  - gender dependent HMMs
  - cluster adaptively trained HMMs (CAT)
  - STC/HLDA models
HTK Discriminative Training Tools: Lattice Implementation

Two sets of model marked lattices required:

- **numerator** lattices: from reference transcription
- **denominator** lattices: from full recognition using weak LM

Efficient lattice level forward-backward algorithm benefits from:

- support of flexible sharing of model parameters
- state and Gaussian level output probability caching
- Gaussian frame occupancy caching
- fixed phone boundary model internal re-alignment - "**Exact Match**"
- batch I/O access of lattices as merged lattice label files (LLF)
Useful common configuration variables:

- **E**: constant used in EBW update, e.g., 2.0
- **LATPROBScale**: acoustic scaling by LM score inverse, e.g., 1/13
- **ISMOOTH**\{TAU,TAUT,TAUW\}: I-smoothing constants, e.g., 50/1/1 for MPE
- **PRIOR**\{TAU,TAUT,TAUW,K\}: static prior, e.g., 25/10/10/1, for MPE-MAP
- **PHONEMEE**: MWE or MPE training
- **EXACTCORRECTNESS**: “Exact” or approximate error in MPE/MWE
- **MMIPRIOR**: use MMI prior
HTK Discriminative Training Tools: 
Supported HTK Features & Limitations

Many other useful HTK features are supported:

- multi-streams, tied-mixtures and parameter tying
- a variety of adaptation schemes, e.g., MMI/MPE-SAT
- lattice based adaptation
- single pass re-train using new front-ends, e.g., bandwidth specific models

Known limitations are:

- only diagonal covariance HMMs supported
- Gaussian means and variances tied on the same level
HTK Discriminative Training Tools: General procedure

Reference transcripts → Word lattices

Uni-gram or heavily pruned bi-gram LM → Word lattices

Speech audio → MLE model → Word lattices

Deterministic lattices → Denominator lattices

MLE model

Numerator lattices → MPE model

HDecode

HMMIMRest

HLRescore

HTKLM
Thank you!