Disclaimer

- The material provided in this document is not my original work and is a summary of some one else’s work(s).
- A simple Google search of the title of the document will direct you to the original source of the material.
- I do not guarantee the accuracy, completeness, timeliness, validity, non-omission, merchantability or fitness of the contents of this document for any particular purpose.
-

Chapter # 7: HMM Definition Files

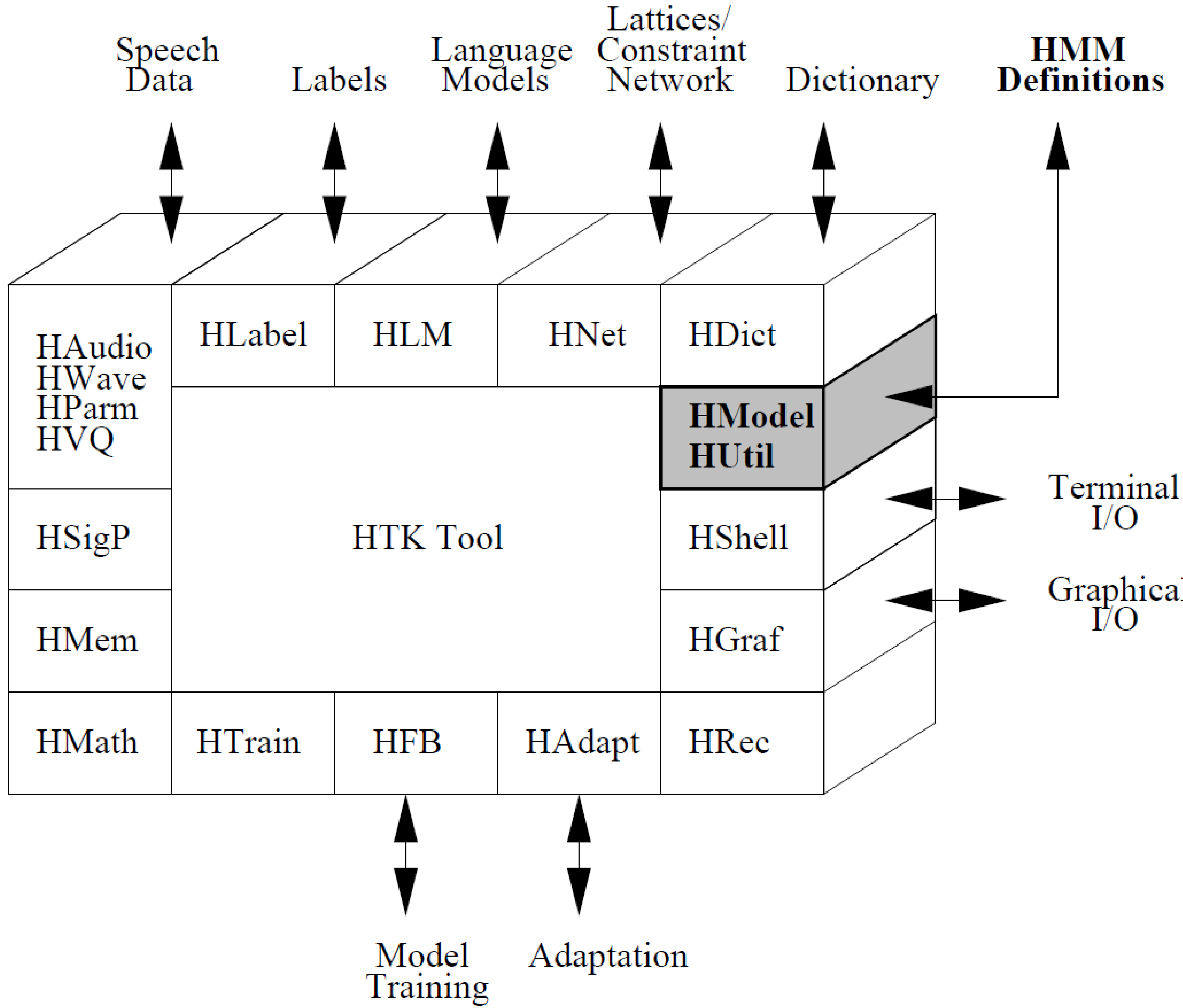
HMM TOOL KIT HTK

Outline

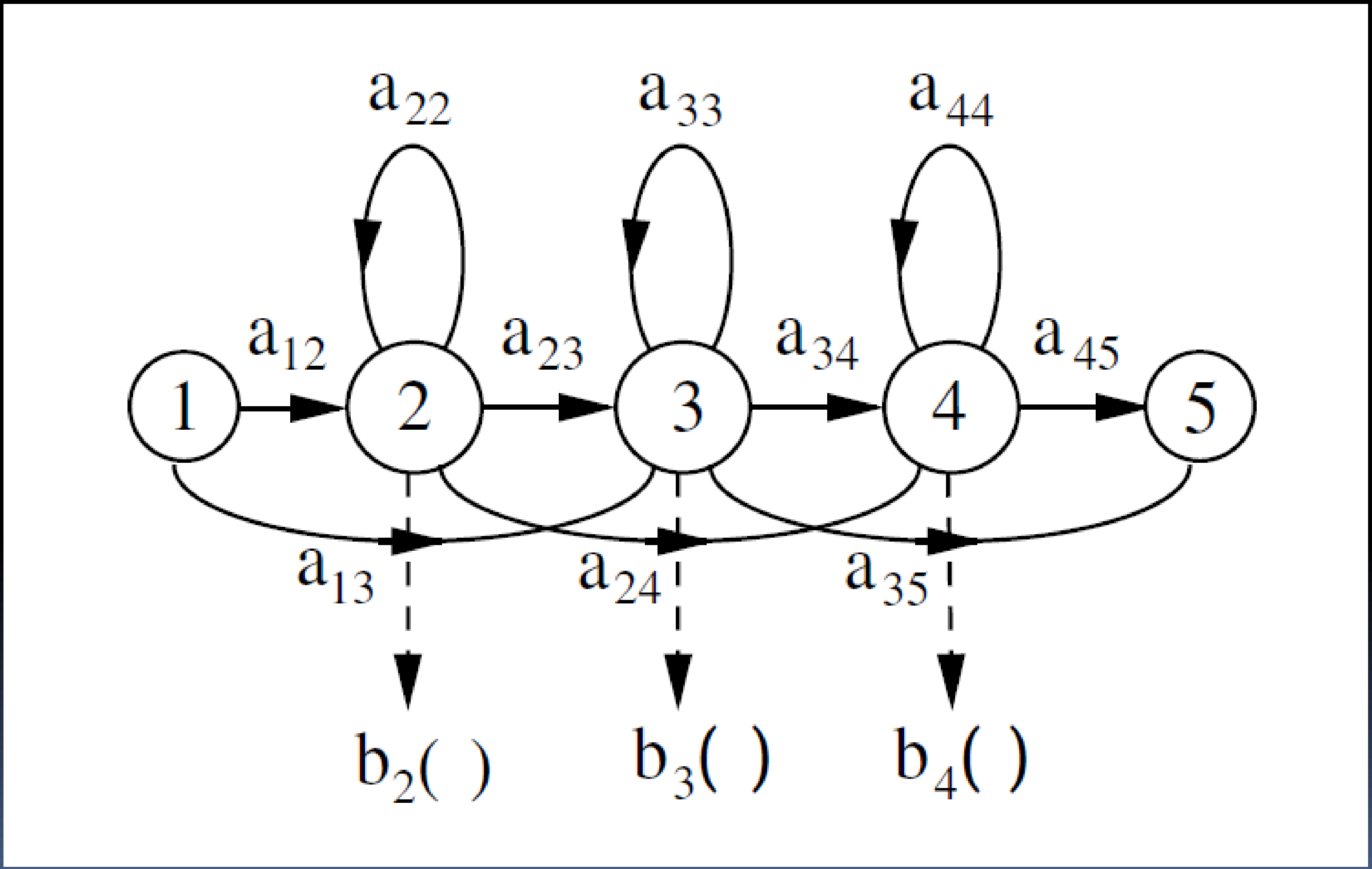
- Introduction
- The HMM Parameters
- Basic HMM Definitions
- Macro Definitions
- HMM Sets
- Tied-Mixture Systems
- Discrete Probability HMMs
- Input Linear Transforms
- Tee Models
- Binary Storage Format
- The HMM Definition Language

Introduction

- The principle function of HTK is to manipulate sets of hidden Markov models (HMMs)
- The definition of a HMM must specify
  - The model topology
  - The transition parameters
  - The output distribution parameters
- The HMM observation vectors can be divided into multiple independent data streams
- HMM can have ancillary information such as duration parameters


Speech Data
Labels
Language Models
Lattices/ Constraint Network
Dictionary
HMM Definitions
HAudio
HWave
HParm
HVQ
HLabel
HLM
HNet
HDict
HModel
HUtil
HSigP
HTK Tool
HShell
Terminal I/O
HMem
HGraf
Graphical I/O
HMath
HTrain
HFB
HAdapt
HRec
Model Training
Adaptation

The HMM Parameters



The HMM Parameters

$$b_j(\boldsymbol{o}_t) = \prod_{s=1}^{S} \left[\sum_{m=1}^{M_{js}} c_{jsm} \mathcal{N}(\boldsymbol{o}_{st}; \boldsymbol{\mu}_{jsm}, \boldsymbol{\Sigma}_{jsm}) \right]^{\gamma_s}$$

$$\mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\boldsymbol{o}-\boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\boldsymbol{o}-\boldsymbol{\mu})}$$

$$b_j(\boldsymbol{o}_t) = \prod_{s=1}^{S} \{P_{js}[v_s(\boldsymbol{o}_{st})]\}^{\gamma_s}$$

The HMM Parameters

- type of observation vector
- number and width of each data stream
- optional model duration parameter vector
- number of states
- for each emitting state and each stream
  - mixture component weights or discrete probabilities
  - if continuous density, then means and covariances
  - optional stream weight vector
  - optional duration parameter vector
- transition matrix

Basic HMM Definitions

```
~h “hmm1”
<BeginHMM>
    <VecSize> 4 <MFCC>
    <NumStates> 5
    <State> 2
        <Mean> 4
            0.2 0.1 0.1 0.9
        <Variance> 4
            1.0 1.0 1.0 1.0
    <State> 3
        <Mean> 4
            0.4 0.9 0.2 0.1
        <Variance> 4
            1.0 2.0 2.0 0.5
    <State> 4
        <Mean> 4
            1.2 3.1 0.5 0.9
        <Variance> 4
            5.0 5.0 5.0 5.0
    <TransP> 5
        0.0 0.5 0.5 0.0 0.0
        0.0 0.4 0.4 0.2 0.0
        0.0 0.0 0.6 0.4 0.0
        0.0 0.0 0.0 0.7 0.3
        0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

Basic HMM

```
~h “hmm2”
<BeginHMM>
    <VecSize> 4 <MFCC>
    <NumStates> 4
    <State> 2 <NumMixes> 2
        <Mixture> 1 0.4
            <Mean> 4
                0.3 0.2 0.2 1.0
            <Variance> 4
                1.0 1.0 1.0 1.0
        <Mixture> 2 0.6
            <Mean> 4
                0.1 0.0 0.0 0.8
            <Variance> 4
                1.0 1.0 1.0 1.0
    <State> 3 <NumMixes> 2
        <Mixture> 1 0.7
            <Mean> 4
                0.1 0.2 0.6 1.4
            <Variance> 4
                1.0 1.0 1.0 1.0
        <Mixture> 2 0.3
            <Mean> 4
                2.1 0.0 1.0 1.8
            <Variance> 4
                1.0 1.0 1.0 1.0
    <TransP> 4
        0.0 1.0 0.0 0.0
        0.0 0.5 0.5 0.0
        0.0 0.0 0.6 0.4
        0.0 0.0 0.0 0.0
<EndHMM>
```

Mixture Gaussian HMM

```
~o <VecSize> 4 <MFCC>
~h “hmm3”
<BeginHMM>
    <NumStates> 4
    <State> 2 <NumMixes> 2
        <Mixture> 1 0.4
            <Mean> 4
                0.3 0.2 0.2 1.0
            <Variance> 4
                1.0 1.0 1.0 1.0
            <Mixture> 2 0.6
            <Mean> 4
                0.1 0.0 0.0 0.8
            <Variance> 4
                1.0 1.0 1.0 1.0
    <State> 3 <NumMixes> 1
        <Mean> 4
            0.10.20.61.4
        <InvCovar> 4
            1.00.10.00.0
                1.00.20.0
                    1.00.1
                        1.0
    <TransP> 4
        0.0 1.0 0.0 0.0
        0.0 0.5 0.5 0.0
        0.0 0.0 0.6 0.4
        0.0 0.0 0.0 0.0
<EndHMM>
```

Full Covarience HMM

Basic HMM Definitions

```
~o  <VecSize> 4 <MFCC>
    <StreamInfo> 2 3 1
~h "hmm4"
<BeginHMM>
    <NumStates> 4
    <State> 2
       <SWeights> 2 0.9 1.1
       <Stream> 1
          <Mean> 3
             0.2 0.1 0.1
          <Variance> 3
             1.0 1.0 1.0
       <Stream> 2
          <Mean> 1 0.0
          <Variance> 1 4.0
    <State> 3
       <Stream> 1
          <Mean> 3
             0.3 0.2 0.0
          <Variance> 3
             1.0 1.0 1.0
       <Stream> 2
          <Mean> 1 0.5
          <Variance> 1 3.0
    <TransP> 4
       0.0 1.0 0.0 0.0
       0.0 0.6 0.4 0.0
       0.0 0.0 0.4 0.6
       0.0 0.0 0.0 0.0
<EndHMM>
```

Multiple Streams

Macro Definitions

- HTK allows the internal parts of a definition to be written as separate units, possibly in several different files, and then referenced by name wherever they are needed
- Such definitions are called macros

Macro Definitions

```
~o <VecSize> 4 <MFCC>

~v "var"
    <Variance> 4
        1.0 1.0 1.0 1.0
```

Macro De

~o <V FCC>

~v “var

<V

1.0

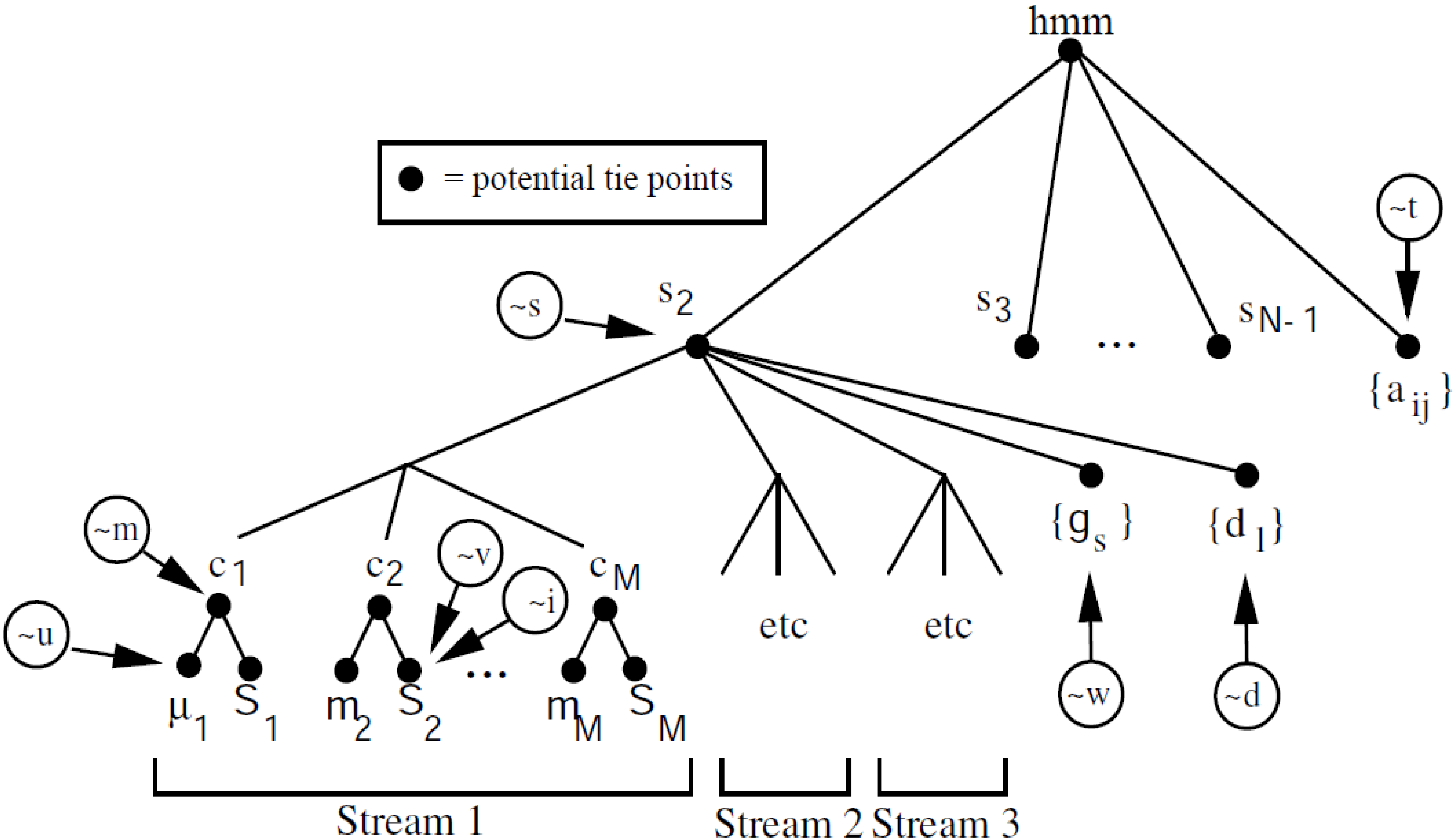
```
~h "hmm5"
<BeginHMM>
    <NumStates> 4
    <State> 2 <NumMixes> 2
        <Mixture> 1 0.4
            <Mean> 4
                0.3 0.2 0.2 1.0
            ~v "var"
        <Mixture> 2 0.6
            <Mean> 4
                0.1 0.0 0.0 0.8
            ~v "var"
    <State> 3 <NumMixes> 2
        <Mixture> 1 0.7
            <Mean> 4
                0.1 0.2 0.6 1.4
            ~v "var"
        <Mixture> 2 0.3
            <Mean> 4
                2.1 0.0 1.0 1.8
            ~v "var"
    <TransP> 4
        0.0 1.0 0.0 0.0
        0.0 0.5 0.5 0.0
        0.0 0.0 0.6 0.4
        0.0 0.0 0.0 0.0
<EndHMM>
```

Using Macro in HMM Definition

Macro Definitions

~s	shared state distribution
~m	shared Gaussian mixture component
~u	shared mean vector
~v	shared diagonal variance vector
~i	shared inverse full covariance matrix
~c	shared choleski L' matrix
~x	shared arbitrary transform matrix
~t	shared transition matrix
~d	shared duration parameters
~w	shared stream weight vector

Macro Definitions



HMM Sets

- Many HTK tools require complete model sets to be specified rather than just a single model
- The individual HMMs which belong to the set are listed in a file rather than being enumerated explicitly on the command line

```
HERest ... -H mf1 -H mf2 ... hlist
```

HMM Sets

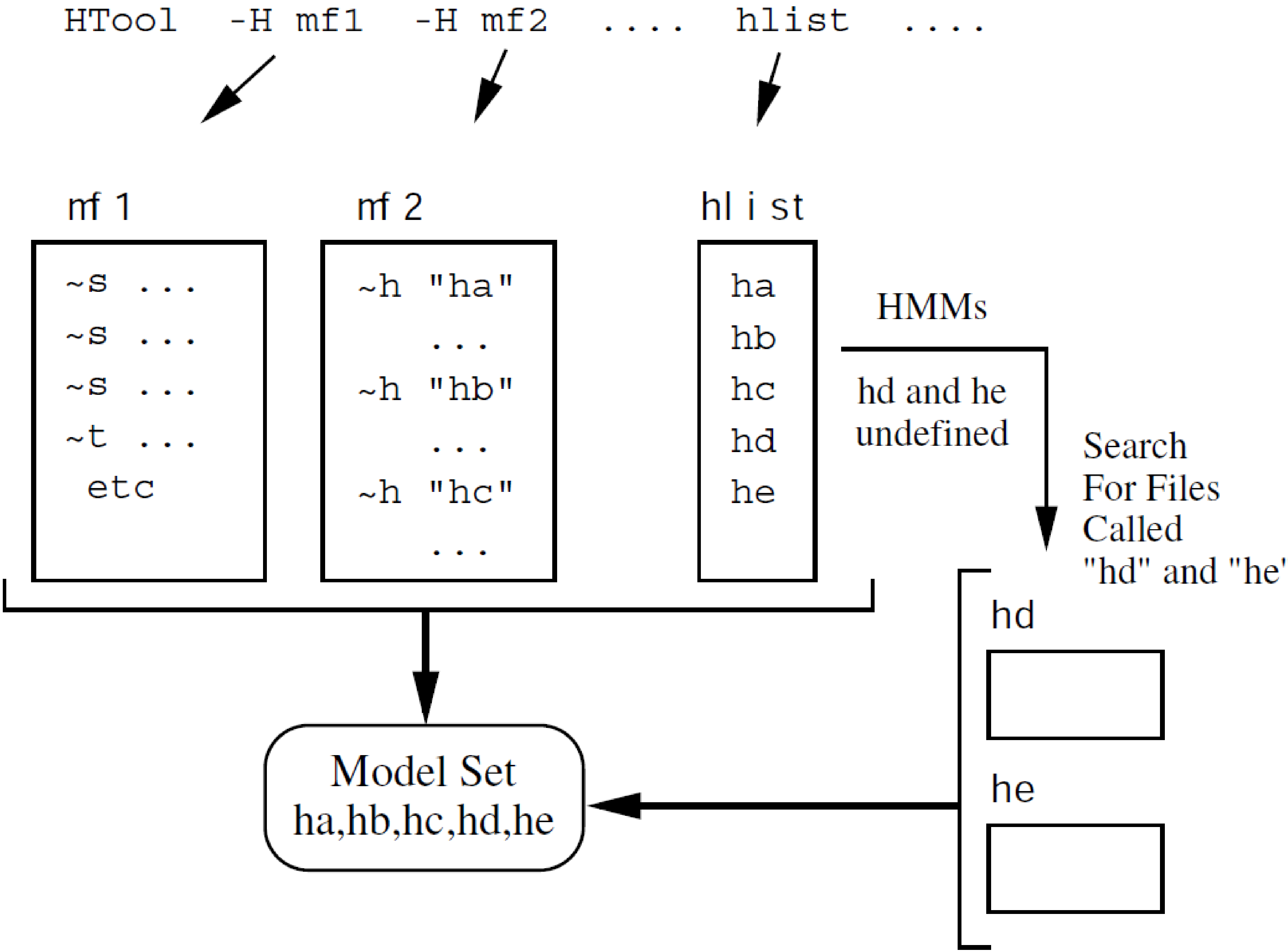
```
~o      <VecSize> 4 <MFCC>
~s "stateA"
    <Mean> 4
       0.2 0.1 0.1 0.9
    <Variance> 4
       1.0 1.0 1.0 1.0
~s "stateB"
    <Mean> 4
       0.4 0.9 0.2 0.1
    <Variance> 4
       1.0 2.0 2.0 0.5
~s "stateC"
    <Mean> 4
       1.2 3.1 0.5 0.9
    <Variance> 4
       5.0 5.0 5.0 5.0
~t "tran"
    <TransP> 5
       0.0 0.5 0.5 0.0 0.0
       0.0 0.4 0.4 0.2 0.0
       0.0 0.0 0.6 0.4 0.0
       0.0 0.0 0.0 0.7 0.3
       0.0 0.0 0.0 0.0 0.0
```

mf1

```
~h "ha"
<BeginHMM>
    <NumStates> 5
    <State> 2
       ~s "stateA"
    <State> 3
       ~s "stateB"
    <State> 4
       ~s "stateB"
    ~t "tran"
<EndHMM>
~h "hb"
<BeginHMM>
    <NumStates> 5
    <State> 2
       ~s "stateB"
    <State> 3
       ~s "stateA"
    <State> 4
       ~s "stateC"
    ~t "tran"
<EndHMM>
```

```
~h "hc"
<BeginHMM>
    <NumStates> 5
    <State> 2
       ~s "stateC"
    <State> 3
       ~s "stateC"
    <State> 4
       ~s "stateB"
    ~t "tran"
<EndHMM>
```

mf2



HTool -H mf1 -H mf2 .... hlist ....
mf1
~s ...
~s ...
~s ...
~t ...
etc
mf2
~h "ha"
...
~h "hb"
...
~h "hc"
...
hlist
ha
hb
hc
hd
he
HMMs
hd and he
undefined
Search
For Files
Called
"hd" and "he"
hd
he
Model Set
ha,hb,hc,hd,he

HMM Sets

- After loading each HMM set, HModel marks it as belonging to one of the following categories called HSKind
  - PLAINHS
  - SHAREDHS
  - TIEDHS
  - DISCRETEHS
- The kind of a HMM set can also be set via the configuration variable HMMSETKIND

HMM Sets

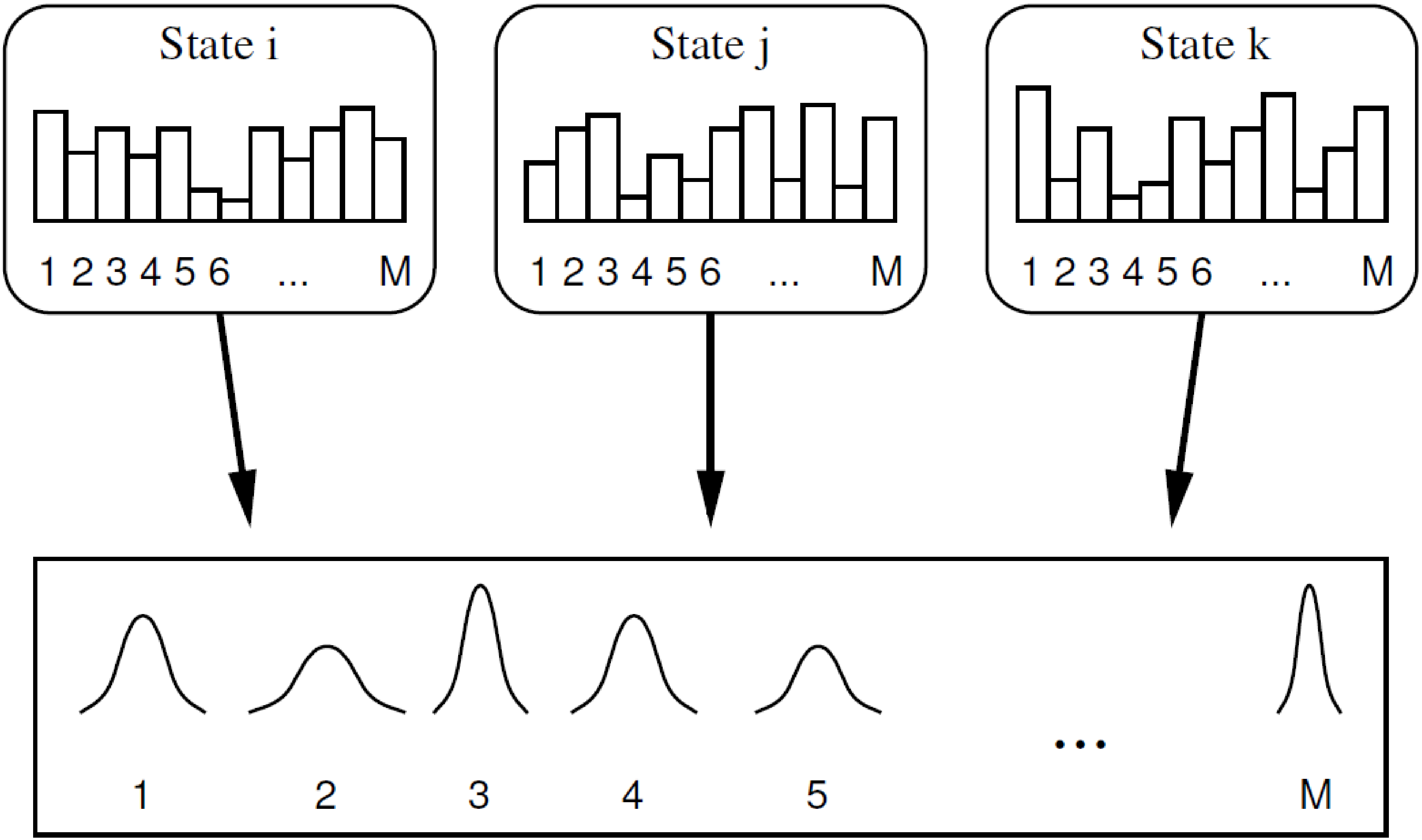
```
two     tuw
too     tuw
to      tuw
one
won     one
three
four
```

HMM List with Tying

Tied-Mixture Systems

- A Tied-Mixture System is one in which all Gaussian components are stored in a pool and all state output distributions share this pool
- Each state output distribution is defined by M mixture component weights
- Since all states share the same components, all of the state-specific discrimination is encapsulated within these weights

Tied-Mixture Systems



Tied-Mixture Codebook

Tied-Mixture Systems

- Each stream then has a separate pool of Gaussians which are often referred to as codebooks
- More formally, for S independent data streams, the output distribution for state j is defined as

$$b_j(\boldsymbol{o}_t) = \prod_{s=1}^{S} \left[\sum_{m=1}^{M_s} c_{jsm} \mathcal{N}(\boldsymbol{o}_{st}; \boldsymbol{\mu}_{sm}, \boldsymbol{\Sigma}_{sm}) \right]^{\gamma_s}$$

Tied-Mixture Systems

```
~o <VecSize> 2 <MFCC>
~m "mix1"
    <Mean> 2 0.0 0.1
    <Variance> 2 1.0 1.0
~m "mix2"
    <Mean> 2 0.2 0.3
    <Variance> 2 2.0 1.0
~m "mix3"
    <Mean> 2 0.0 0.1
    <Variance> 2 1.0 2.0
~m "mix4"
    <Mean> 2 0.4 0.1
    <Variance> 2 1.0 1.5
~m "mix5"
    <Mean> 2 0.9 0.7
    <Variance> 2 1.5 1.0
```

```
~h "htm"
<BeginHMM>
    <NumStates> 4
    <State> 2 <NumMixes> 5
        <TMix> mix 0.2 0.1 0.3 0.3 0.1
    <State> 3 <NumMixes> 5
        <TMix> mix 0.4 0.3 0.1 0.1 0.1
    <TransP> 4
        0.0 1.0 0.0 0.0
        0.0 0.5 0.5 0.0
        0.0 0.0 0.6 0.4
        0.0 0.0 0.0 0.0
<EndHMM>
```

Tied-Mixture Systems

```
~o <VecSize> 2 <MFCC>
~m "mix1"
    <Mean> 2 0.0 0.1
    <Variance> 2 1.0 1.0
~m "mix2"
    <Mean> 2 0.2 0.3
    <Variance> 2 2.0 1.0
~m "mix3"
    <Mean> 2 0.0 0.1
    <Variance> 2 1.0 2.0
~m "mix4"
    <Mean> 2 0.4 0.1
    <Variance> 2 1.0 1.5
~m "mix5"
    <Mean> 2 0.9 0.7
    <Variance> 2 1.5 1.0
```

```
~h "htm"
<BeginHMM>
    <NumStates> 4
    <State> 2 <NumMixes> 5
        <TMix> mix 0.2 0.1 0.3*2 0.1
    <State> 3 <NumMixes> 5
        <TMix> mix 0.4 0.3 0.1*3
    <TransP> 4
        ...
<EndHMM>
```

Discrete Probability HMMs

- Discrete probability HMMs model observation sequences which consist of symbols drawn from a discrete and finite set of size M

Disc... ...Ms

- Dis... model obs... nsist of syn... finite set of s...

```
~o <DISCRETE> <StreamInfo> 2 1 1
~h "dhmm1"
<BeginHMM>
    <NumStates> 4
    <State> 2
        <NumMixes> 10 2
        <SWeights> 2 0.9 1.1
        <Stream> 1
            <DProb> 3288*4 32767*6
        <Stream> 2
            <DProb> 1644*2
    <State> 3
        <NumMixes> 10 2
        <SWeights> 2 0.9 1.1
        <Stream> 1
            <DProb> 5461*10
        <Stream> 2
            <DProb> 1644*2
    <TransP> 4
        0.0 1.0 0.0 0.0
        0.0 0.5 0.5 0.0
        0.0 0.0 0.6 0.4
        0.0 0.0 0.0 0.0
<EndHMM>
```

Input Linear Transforms

```
~j "lintran.mat"
<MMFIdMask> *
<MFCC>
<PreQual>
<LinXform>
    <VecSize> 2
    <BlockInfo> 1 2
    <Block> 1
        <Xform> 2 5
            1.0 0.1 0.2 0.1 0.4
            0.2 1.0 0.1 0.1 0.1
```

Tee Models

- Models which have a non-zero entry to exit transition probability are referred to as tee models
- Tee-models are useful for modelling optional transient effects such as short pauses and noise bursts, particularly between words
- Tee-models are incompatible with those that work with isolated models such as HInit and Hrest

Binary Storage Format

- For experimental work, text-based storage allows simple and direct access to HMM parameters
- When using very large HMM sets, storage in text form is less practical since it is inefficient in its use of memory and the time taken to load can be excessive due to the large number of character to float conversions needed
- HTK tools provide a standard command line option (-B) to indicate that HMM definitions should be output in binary format

The HMM Definition Language

hmmdef = [~h macro]
<BeginHMM>
    [globalOpts]
    <NumStates> short
    state { state }
    transP
    [duration]
<EndHMM>

globalOpts = option { option }
option = <HmmSetId> string |
<StreamInfo> short { short } |
<VecSize> short |
<ProjSize> short |
<InputXform> inputXform |
<ParentXform> ~a macro |
covkind |
durkind |
parmkind

Phoneme Based Recognition

gram - Notepad

File Edit Format View Help

```
$digit = ONE | TWO | THREE | FOUR | FIVE | SIX | SEVEN | EIGHT | NINE | ZERO ;
(SENT-START(<$digit>)SENT-END)
```

Phoneme Based Recognition

words.mlf - WordPad

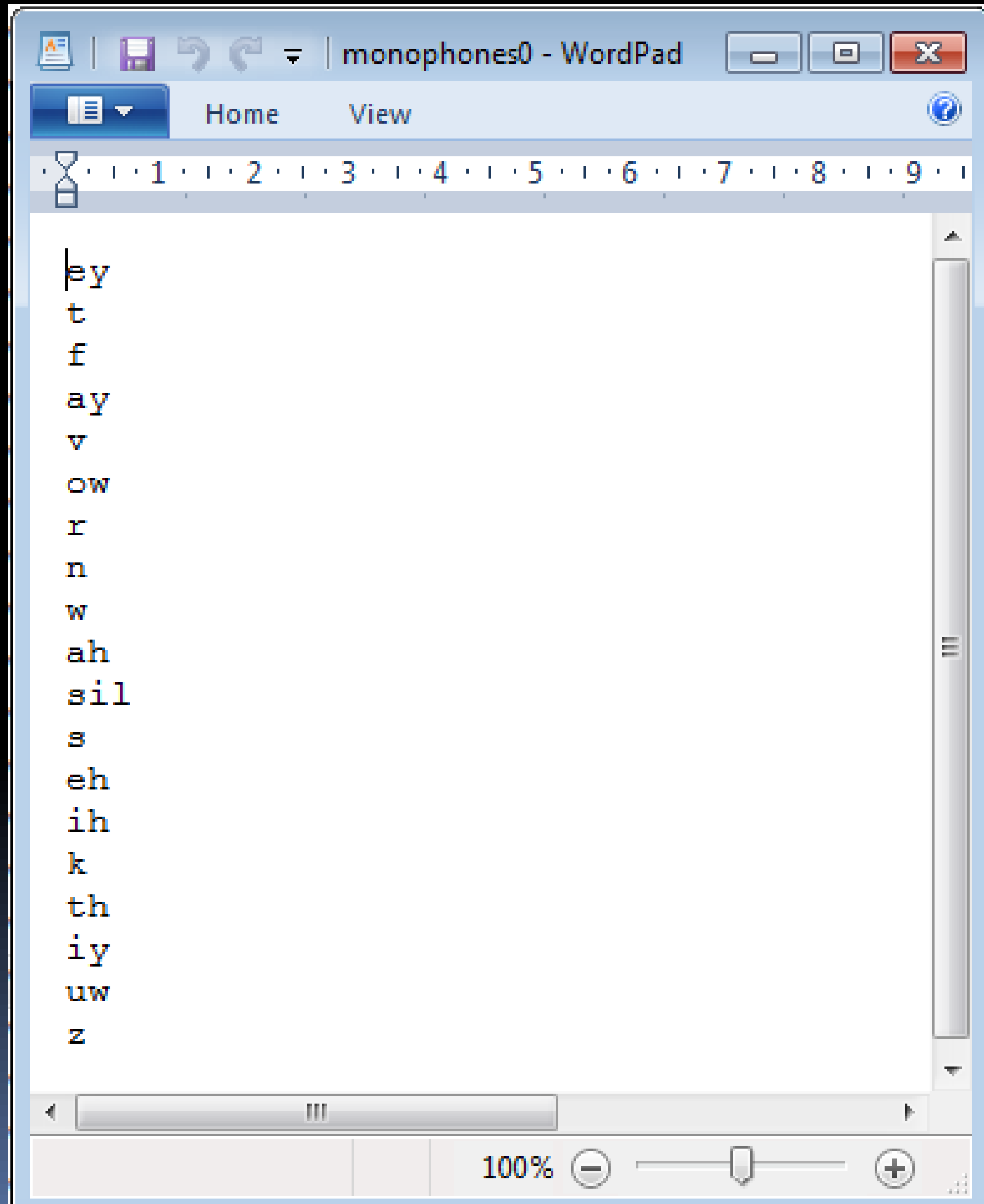
```
#!MLF!#
"*/1-*.lab"
ONE
.
"*/2-*.lab"
TWO
.
"*/3-*.lab"
THREE
.
"*/4-*.lab"
FOUR
.
"*/5-*.lab"
FIVE
.
"*/6-*.lab"
SIX
.
"*/7-*.lab"
SEVEN
.
"*/8-*.lab"
EIGHT
.
"*/9-*.lab"
NINE
.
"*/0-*.lab"
ZERO
.
```

100%

Phoneme Based Recognition

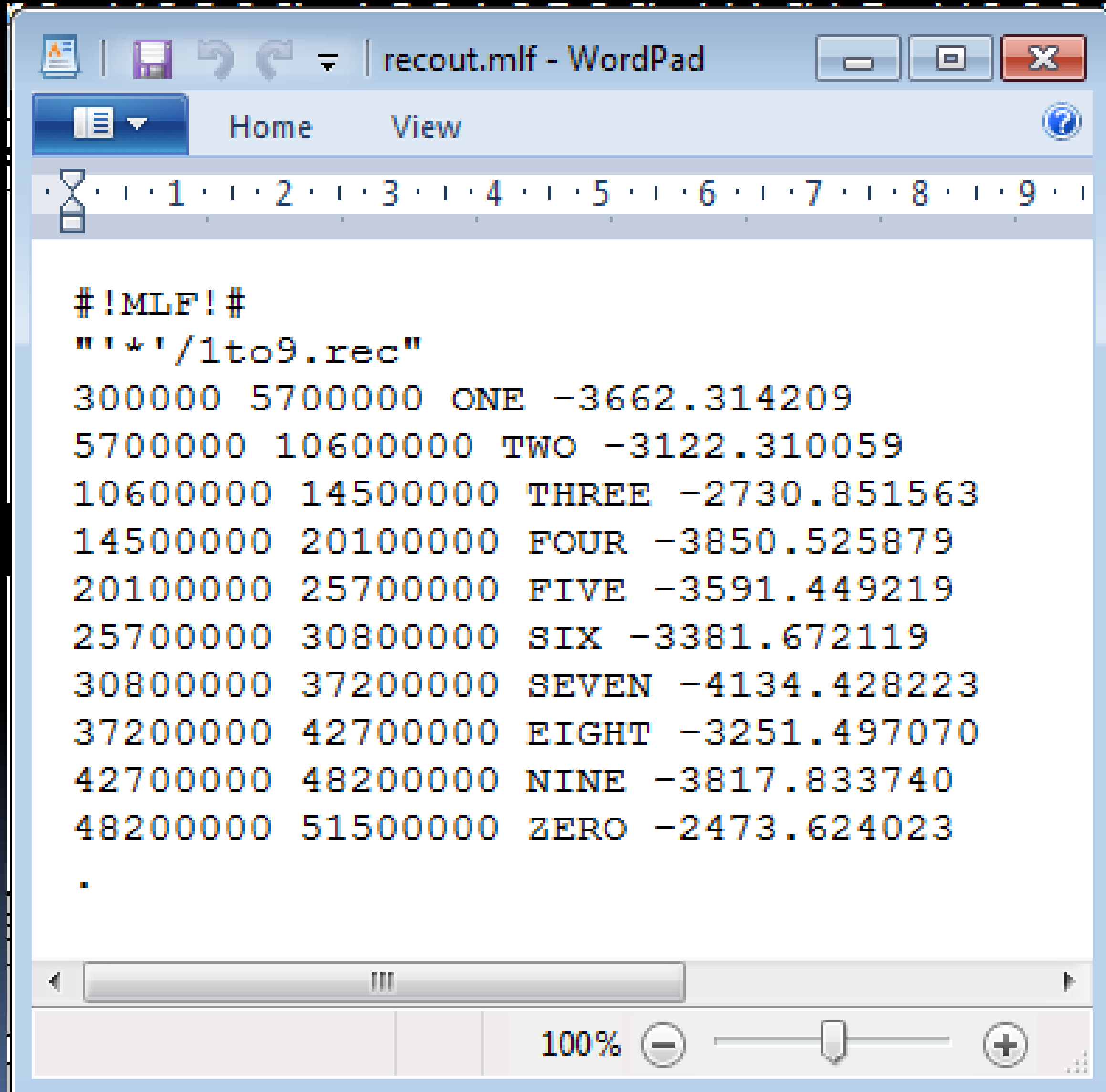
EIGHT	[EIGHT]	ey t sp
FIVE	[FIVE]	f ay v sp
FOUR	[FOUR]	f ow r sp
NINE	[NINE]	n ay n sp
ONE	[ONE]	w ah n sp
SENT-END	[]	sil
SENT-START	[]	sil
SEVEN	[SEVEN]	s eh v ih n sp
SIX	[SIX]	s ih k s sp
THREE	[THREE]	th r iy sp
TWO	[TWO]	t uw sp
ZERO	[ZERO]	z iy r ow sp

Phoneme Based Recognition



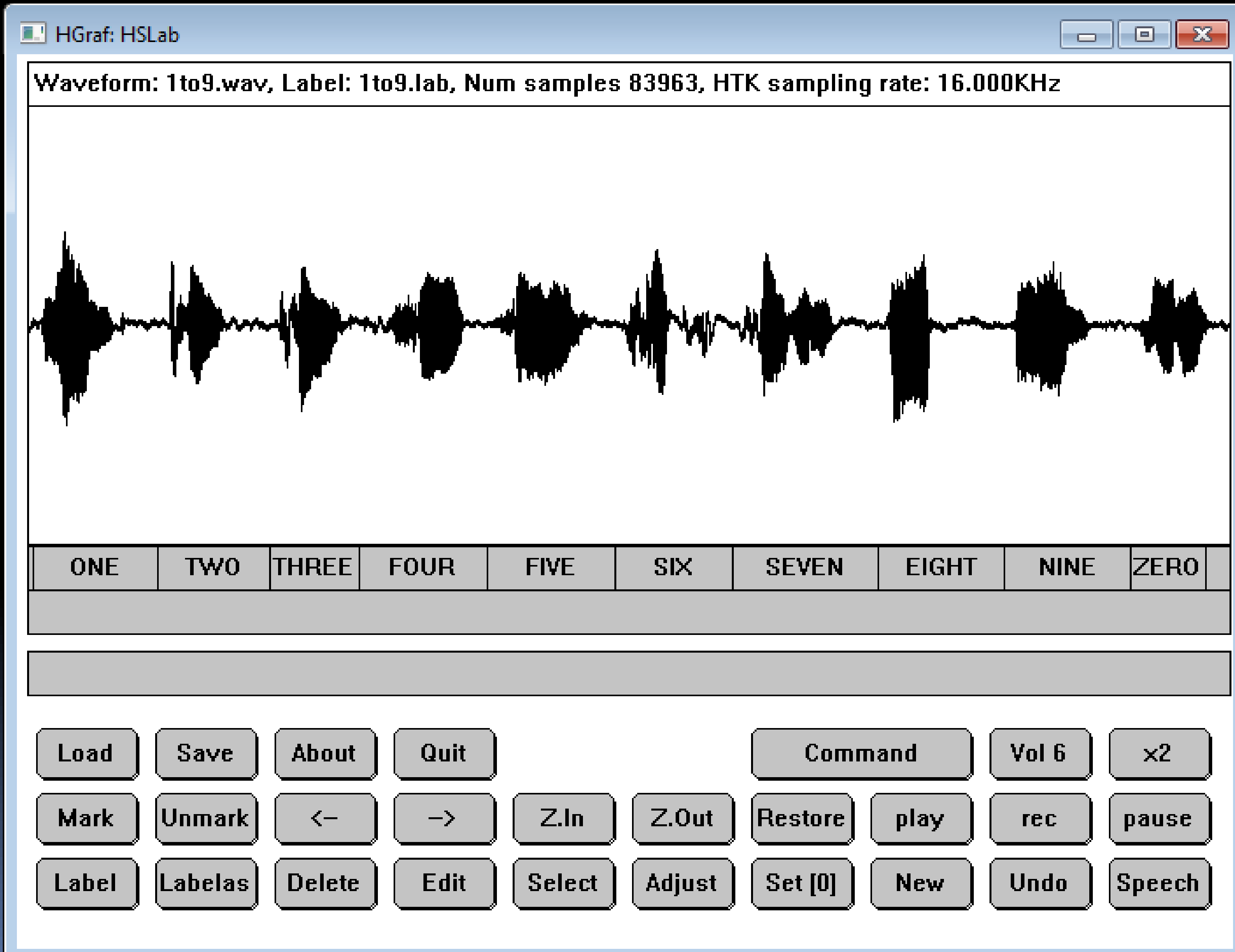
ey
t
f
ay
v
ow
r
n
w
ah
sil
s
eh
ih
k
th
iy
uw
z

Phoneme Based Recognition




```
#!MLF!#
"'*'/1to9.rec"
300000 5700000 ONE -3662.314209
5700000 10600000 TWO -3122.310059
10600000 14500000 THREE -2730.851563
14500000 20100000 FOUR -3850.525879
20100000 25700000 FIVE -3591.449219
25700000 30800000 SIX -3381.672119
30800000 37200000 SEVEN -4134.428223
37200000 42700000 EIGHT -3251.497070
42700000 48200000 NINE -3817.833740
48200000 51500000 ZERO -2473.624023
.
```

Phoneme Based Recognition



Phoneme Based Recognition

```
READY[1]>
Please speak sentence - measuring levels
Level measurement completed
SENT-START ONE TWO THREE SENT-END  ==  [196 frames] -62.2453 [Ac=-12200.1 LM=0.0] (Act=56.9)

READY[2]>
SENT-START FOUR FIVE SIX SENT-END  ==  [220 frames] -64.3957 [Ac=-14167.1 LM=0.0] (Act=57.2)

READY[3]>
SENT-START SEVEN EIGHT NINE SENT-END  ==  [211 frames] -67.4605 [Ac=-14234.2 LM=0.0] (Act=57.1)

READY[4]>
SENT-START NINE EIGHT SENT-END  ==  [173 frames] -61.5619 [Ac=-10650.2 LM=0.0] (Act=56.7)

READY[5]>
SENT-START SEVEN SIX FIVE SENT-END  ==  [192 frames] -68.6498 [Ac=-13180.8 LM=0.0] (Act=56.9)

READY[6]>
SENT-START FOUR THREE TWO SENT-END  ==  [175 frames] -64.4076 [Ac=-11271.3 LM=0.0] (Act=56.7)

READY[7]>
SENT-START ZERO SENT-END  ==  [118 frames] -62.5132 [Ac=-7376.6 LM=0.0] (Act=55.6)
```

Thank You