HTK Tutorial

Giampiero Salvi

KTH (Royal Institute of Technology),
Dep. of Speech, Music and Hearing,
Drottning Kristinas v. 31,
SE-100 44, Stockholm, Sweden
giampi@kth.se

Nov. 2003

Introduction

Data formats and manipulation

Data visualization

Training

Recognition

HTK, What is it?

- A toolkit for Hidden Markov Modeling

HTK, What is it?

- A toolkit for Hidden Markov Modeling
- General purpose, but...

HTK, What is it?

- A toolkit for Hidden Markov Modeling
- General purpose, but...
- ...optimized for Speech Recognition

HTK, What is it?

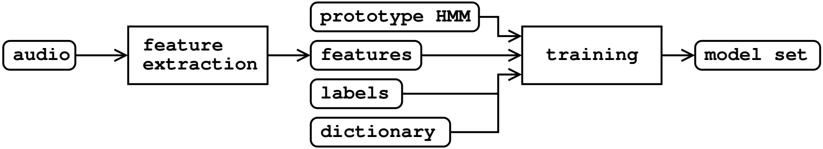
- A toolkit for Hidden Markov Modeling
- General purpose, but...
- ...optimized for Speech Recognition
- Very flexible and complete (active development)

HTK, What is it?

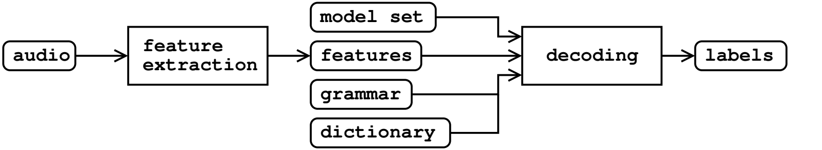
- A toolkit for Hidden Markov Modeling
- General purpose, but...
- ...optimized for Speech Recognition
- Very flexible and complete (active development)
- Very good documentation (HTKBook)

ASR Overview

Training


audio
feature extraction
prototype HMM
features
labels
dictionary
training
model set

Recognition


audio
feature extraction
model set
features
grammar
dictionary
decoding
labels

Things that you should have before you start

- familiarity with Unix-like shell
  - `cd, ls, pwd, mkdir, cp, foreach...`

Things that you should have before you start

- familiarity with Unix-like shell
  - `cd, ls, pwd, mkdir, cp, foreach...`
- text processing tools:
  - `perl, perl, perl, perl, perl`
  - `grep, gawk, tr, sed, find, cat, wc...`

Things that you should have before you start

- familiarity with Unix-like shell
  - `cd, ls, pwd, mkdir, cp, foreach...`
- text processing tools:
  - `perl, perl, perl, perl, perl`
  - `grep, gawk, tr, sed, find, cat, wc...`
- lots of patience

Things that you should have before you start

- familiarity with Unix-like shell
  - `cd, ls, pwd, mkdir, cp, foreach...`
- text processing tools:
  - `perl, perl, perl, perl, perl`
  - `grep, gawk, tr, sed, find, cat, wc...`
- lots of patience
- the fabulous **HTK Book**

Things that you should have before you start

- familiarity with Unix-like shell
  - `cd, ls, pwd, mkdir, cp, foreach...`
- text processing tools:
  - `perl, perl, perl, perl, perl`
  - `grep, gawk, tr, sed, find, cat, wc...`
- lots of patience
- the fabulous **HTK Book**
- a look at the **RefRec** scripts

The HTK tools

- data manipulation tools:
  `HCopy` `HQuant` `HLEd` `HHEd` `HDMan` `HBuild`

The HTK tools

- data manipulation tools:
  `HCopy` `HQuant` `HLEd` `HHEd` `HDMan` `HBuild`
- data visualization tools:
  `HSLab` `HList` `HSGen`

The HTK tools

- data manipulation tools:
  `HCopy HQuant HLEd HHEd HDMan HBuild`
- data visualization tools:
  `HSLab HList HSGen`
- training tools:
  `HCompV HInit HRest HERest HEAdapt HSmooth`

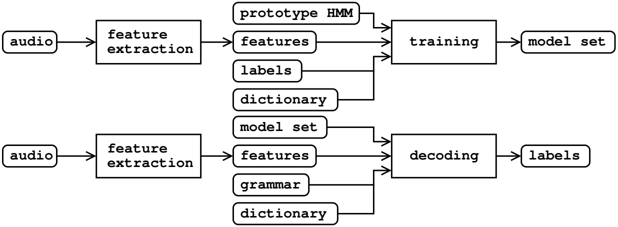
The HTK tools

- data manipulation tools:
  HCopy HQuant HLEd HHEd HDMan HBuild
- data visualization tools:
  HSLab HList HSGen
- training tools:
  HCompV HInit HRest HERest HEAdapt HSmooth
- recognition tools:
  HLStats HParse HVite HResults

The HTK data formats

data formats:

audio:	many common formats plus HTK	binary
features:	HTK	binary
labels:	HTK (single or *Master Label* files)	text
models:	HTK (single or *Master Macro* files)	text or binary
other:	HTK	text



Usage example (HList)

```
> HList

USAGE: HList [options] file ...

 Option                                        Default

 -d      Coerce observation to VQ symbols      off
 -e N    End at sample N                       0
 -h      Print source header info              off
 -i N    Set items per line to N               10
 -n N    Set num streams to N                  1
 -o      Print observation structure           off
 -p      Playback audio                        off
 -r      Write raw output                      off
 -s N    Start at sample N                     0
 -t      Print target header info              off
 -z      Suppress printing data                on
 -A      Print command line arguments          off
 -C cf   Set config file to cf                 default
 -D      Display configuration variables       off
...
```

Command line switches and options

```
> HList -e 1 -o -h feature_file

 Source: feature_file
  Sample Bytes:  26       Sample Kind:   MFCC_0
  Num Comps:     13       Sample Period: 10000.0 us
  Num Samples:   336      File Format:   HTK
-------------------- Observation Structure ---------------------
x:      MFCC-1  MFCC-2  MFCC-3  MFCC-4  MFCC-5  MFCC-6  MFCC-7
        MFCC-8  MFCC-9 MFCC-10 MFCC-11 MFCC-12      C0
------------------------ Samples: 0->1 -------------------------
0:     -14.314  -3.318  -6.263  -7.245   7.192   4.997   0.830
         3.293   5.428   6.831   5.819   5.606  40.734
1:     -13.591  -4.756  -6.037  -3.362   3.541   3.510   2.867
         0.812   0.630   5.285   1.054   8.375  40.778
----------------------------- END ------------------------------
```

Configuration file

```
> cat config_file

SOURCEKIND = MFCC_0
TARGETKIND = MFCC_0_D_A

> HList -C config_file -e 0 -o -h feature_file

 Source: feature_file
  Sample Bytes:  26       Sample Kind:   MFCC_0
  Num Comps:     13       Sample Period: 10000.0 us
  Num Samples:   336      File Format:   HTK
-------------------- Observation Structure ---------------------
x:      MFCC-1   MFCC-2   MFCC-3   MFCC-4   MFCC-5   MFCC-6   MFCC-7
        MFCC-8   MFCC-9  MFCC-10  MFCC-11  MFCC-12       C0    Del-1
         Del-2    Del-3    Del-4    Del-5    Del-6    Del-7    Del-8
         Del-9   Del-10   Del-11   Del-12    DelC0    Acc-1    Acc-2
         Acc-3    Acc-4    Acc-5    Acc-6    Acc-7    Acc-8    Acc-9
        Acc-10   Acc-11   Acc-12    AccC0
------------------------ Samples: 0->1 -------------------------
0:     -14.314   -3.318   -6.263   -7.245    7.192    4.997    0.830
         3.293    5.428    6.831    5.819    5.606   40.734   -0.107
        -0.180    0.731    1.134   -0.723   -0.676    1.083   -0.552
        -0.387   -0.592   -2.172   -0.030   -0.170    0.236    0.170
        -0.241   -0.226   -0.517   -0.244   -0.053    0.213   -0.029
         0.097    0.225   -0.294    0.051
----------------------------- END ------------------------------
```

File manipulation tools

- `HCopy`: converts from/to various data formats (audio, **features**).
- `HQuant`: quantizes speech (audio).
- `HLEd`: edits label and **master label files**.
- `HDMan`: edits **dictionary files**.
- `HHEd`: edits model and **master macro files**.
- `HBuild`: converts language models in different formats (more in recognition section).

Computing feature files (HCopy)

```
> cat config_file

# Feature configuration
TARGETKIND = MFCC_0
TARGETRATE = 100000.0
SAVECOMPRESSED = T
SAVEWITHCRC = T
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 26
CEPLIFTER = 22
NUMCEPS = 12
ENORMALISE = F
# input file format (headerless 8 kHz 16 bit linear PCM)
SOURCEKIND = WAVEFORM
SOURCEFORMAT = NOHEAD
SOURCERATE = 1250

> HCopy -C config_file audio_file1 param_file1 audio_file2 ...

> HCopy -C config_file -S file_list
```

Label files

```
#!MLF!#
"filename1"
 [start1 [end1]]    label1 [score]    {auxlabel [auxscore]}    [comment]
 [start2 [end2]]    label2 [score]    {auxlabel [auxscore]}    [comment]
 ...
 [startN [endN]]    labelN [score]    {auxlabel [auxscore]}    [comment]
.
"filename2"
 ...
.
```

- [.] = optional (0 or 1);
- {.} = possible repetition (0, 1, 2...)
- time stamps are in 100ns units (!?): 10ms = 100.000

Label file example 1

```
> cat aligned.mlf

#!MLF!#
"*/a10001a1.rec"
       0  6400000 sil    <sil>
 6400000  8600000 f      förra
 8600000 10400000 oe
10400000 11700000 r
11700000 14100000 a
14100000 14100000 sp
14100000 29800001 sil    <sil>
.
"*/a10001i1.rec"
       0  2600000 sil    <sil>
 2600000  4900000 S      sju
 4900000  8300000 uh:
 8300000  8600000 a
 8600000  8600000 sp
 8600000 21600000 sil    <sil>
.
```

Label file example 2 (HLEd)

```
> HLEd -l ’*’ -d lex.dic -i phones.mlf words2phones.led words.mlf
```

```
> cat words.mlf

#!MLF!#
"*/a10001a1.rec"
förra
.
"*/a10001i1.rec"
sju
.
```

```
> cat words2phones.led

EX
IS sil sil
```

```
> cat phones.mlf

#!MLF!#
"*/a10001a1.rec"
sil
f
oe
r
a
sp
sil
.
"*/a10001i1.rec"
sil
S
uh:
a
sp
sil
.
```

Dictionary (HDMan)

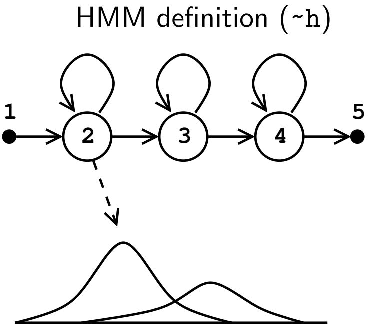
```
WORD [OUTSYM] PRONPROB P1 P2 P3 P4 ...
```

```
> cat lex.dic

förra   f oe r a sp
sju     S uh: a sp
```

```
> cat lex2.dic

<sil>   [] sil
förra   f oe r a sp
sju     0.3 S uh: a sp
sju     0.7 S uh: sp
```

HMM definition files (HHEd)

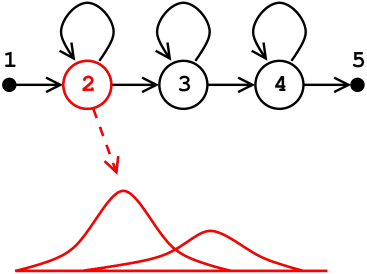
```
~h "hmm_name"
<BEGINHMM>
  <NUMSTATES> 5
  <STATE> 2
    <NUMMIXES> 2
    <MIXTURE> 1 0.8
      <MEAN> 4
        0.1 0.0 0.7 0.3
      <VARIANCE> 4
        0.2 0.1 0.1 0.1
    <MIXTURE> 2 0.2
      <MEAN> 4
        0.2 0.3 0.4 0.0
      <VARIANCE> 4
        0.1 0.1 0.1 0.2
  <STATE> 3
    ~s "state_name"
  <STATE> 4
    <NUMMIXES> 2
    <MIXTURE> 1 0.7
      ~m "mix_name"
    <MIXTURE> 2 0.3
      <MEAN> 4
        ~u "mean_name"
      <VARIANCE> 4
        ~v "variance_name"
  <TRANSP>
    ~t "transition_name"
<ENDHMM>
```



HMM definition files (HHEd)

```
~h "hmm_name"
<BEGINHMM>
  <NUMSTATES> 5
  <STATE> 2
    <NUMMIXES> 2
    <MIXTURE> 1 0.8
      <MEAN> 4
        0.1 0.0 0.7 0.3
      <VARIANCE> 4
        0.2 0.1 0.1 0.1
    <MIXTURE> 2 0.2
      <MEAN> 4
        0.2 0.3 0.4 0.0
      <VARIANCE> 4
        0.1 0.1 0.1 0.2
  <STATE> 3
    ~s "state_name"
  <STATE> 4
    <NUMMIXES> 2
    <MIXTURE> 1 0.7
      ~m "mix_name"
    <MIXTURE> 2 0.3
      <MEAN> 4
        ~u "mean_name"
      <VARIANCE> 4
        ~v "variance_name"
  <TRANSP>
    ~t "transition_name"
<ENDHMM>
```

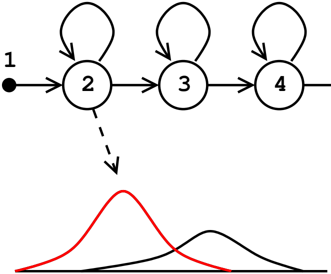
State definition (~s)



HMM definition files (HHEd)

```
~h "hmm_name"
<BEGINHMM>
  <NUMSTATES> 5
  <STATE> 2
    <NUMMIXES> 2
    <MIXTURE> 1 0.8
      <MEAN> 4
        0.1 0.0 0.7 0.3
      <VARIANCE> 4
        0.2 0.1 0.1 0.1
    <MIXTURE> 2 0.2
      <MEAN> 4
        0.2 0.3 0.4 0.0
      <VARIANCE> 4
        0.1 0.1 0.1 0.2
  <STATE> 3
    ~s "state_name"
  <STATE> 4
    <NUMMIXES> 2
    <MIXTURE> 1 0.7
      ~m "mix_name"
    <MIXTURE> 2 0.3
      <MEAN> 4
        ~u "mean_name"
      <VARIANCE> 4
        ~v "variance_name"
  <TRANSP>
    ~t "transition_name"
<ENDHMM>
```

Gaussian mixture component definition (~m)

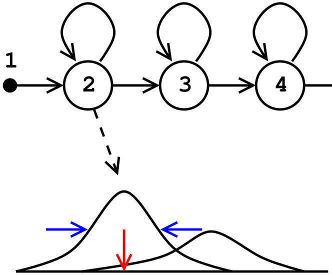


HMM definition files (HHEd)

```
~h "hmm_name"
<BEGINHMM>
  <NUMSTATES> 5
  <STATE> 2
    <NUMMIXES> 2
    <MIXTURE> 1 0.8
      <MEAN> 4
        0.1 0.0 0.7 0.3
      <VARIANCE> 4
        0.2 0.1 0.1 0.1
    <MIXTURE> 2 0.2
      <MEAN> 4
        0.2 0.3 0.4 0.0
      <VARIANCE> 4
        0.1 0.1 0.1 0.2
  <STATE> 3
    ~s "state_name"
  <STATE> 4
    <NUMMIXES> 2
    <MIXTURE> 1 0.7
      ~m "mix_name"
    <MIXTURE> 2 0.3
      <MEAN> 4
        ~u "mean_name"
      <VARIANCE> 4
        ~v "variance_name"
  <TRANSP>
    ~t "transition_name"
<ENDHMM>
```

Mean vector definition (~u)

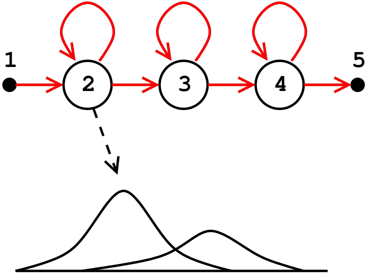
Diagonal variance vector definition (~v)



HMM definition files (HHEd)

```
~h "hmm_name"
<BEGINHMM>
  <NUMSTATES> 5
  <STATE> 2
    <NUMMIXES> 2
    <MIXTURE> 1 0.8
      <MEAN> 4
        0.1 0.0 0.7 0.3
      <VARIANCE> 4
        0.2 0.1 0.1 0.1
    <MIXTURE> 2 0.2
      <MEAN> 4
        0.2 0.3 0.4 0.0
      <VARIANCE> 4
        0.1 0.1 0.1 0.2
  <STATE> 3
    ~s "state_name"
  <STATE> 4
    <NUMMIXES> 2
    <MIXTURE> 1 0.7
      ~m "mix_name"
    <MIXTURE> 2 0.3
      <MEAN> 4
        ~u "mean_name"
      <VARIANCE> 4
        ~v "variance_name"
  <TRANSP>
    ~t "transition_name"
<ENDHMM>
```

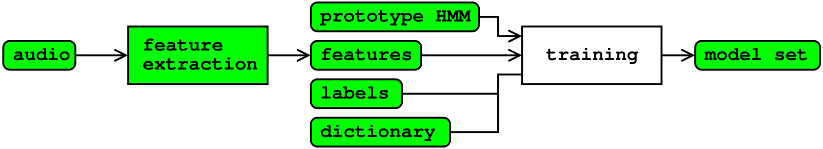
Transition matrix definition (~t)



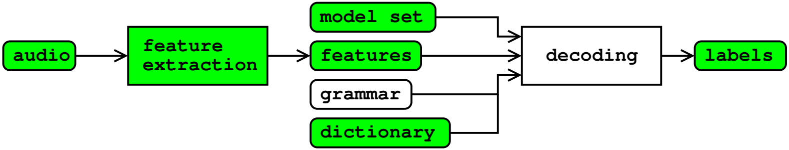
- HSLab: graphical tool to label speech (use WaveSurfer instead).
- HList: gives information about audio and feature files.
- HSGen: generates random sentences out of a regular grammar.

Intermezzo: what do we know so far?

Training



Recognition



model initialization

Initialization procedure depends on the information avaliable at that time.

- `HCompV`: computes the overall mean and variance.
  Input: a prototype HMM.
- `HInit`: Viterbi segmentation + parameter estimation. For mixture distribution uses K-means.

  Input: a prototype HMM, time aligned transcriptions.

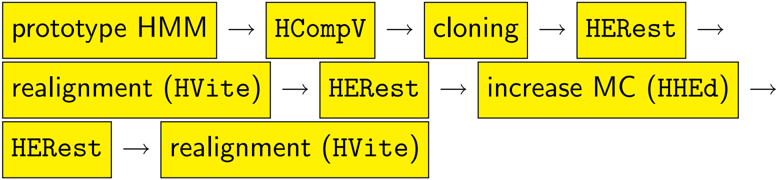
Traning tools

- `HRest`: Baum-Welch re-estimation.

  Input: an initialized model set, time aligned transcriptions.
- `HERest`: performs *embedded* Baum-Welch training.
  Input: an initialized model set, timeless transcriptions.
- `HEAdapt`: performs adaptation on a limited set of data.
- `HSmooth`: smoots a set of context-dependent models according to the context-independent counterpart.

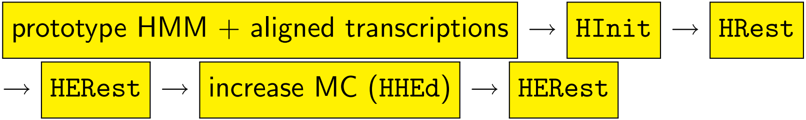
Training example: RefRec

first pass:



prototype HMM → HCompV → cloning → HERest → realignment (HVite) → HERest → increase MC (HHEd) → HERest → realignment (HVite)

second pass:



prototype HMM + aligned transcriptions → HInit → HRest → HERest → increase MC (HHEd) → HERest

Recognition tools

grammar generation

- `HLStats`: creates bigram from training data.
- `HParse`: parses a user defined grammar to produce a *lattice*.

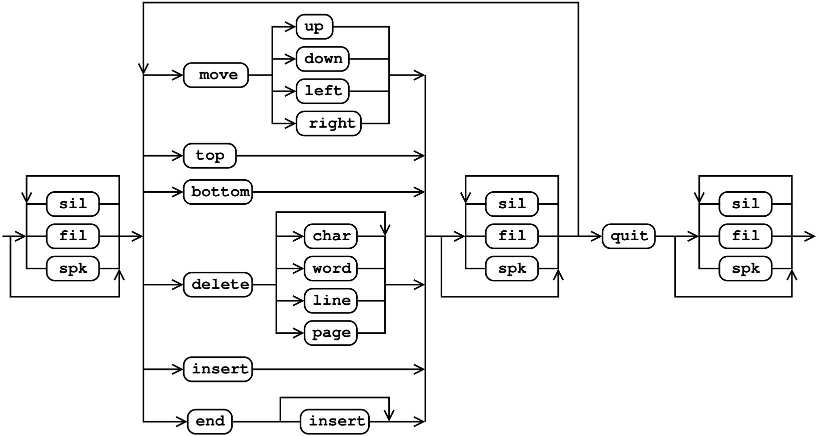
decoding

- `HVite`: performs Viterbi decoding.

evaluation

- `HResults`: evaluates recognition results.

Grammar definition (HParse)



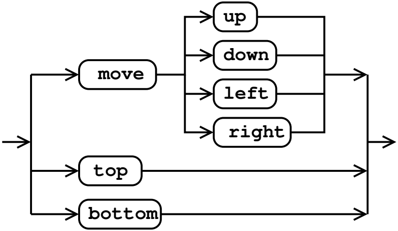
Grammar definition (HParse)

```
> cat grammar.bnf
$dir  = up | down | left | right;
$mcmd = move $dir | top | bottom;
$item = char | word | line | page;
$dcmd = delete [$item];
$icmd = insert;
$ecmd = end [insert];
$cmd  = $mcmd | $dcmd | $icmd | $ecmd;
$noise     = sil | fil | spk;
({$noise} < $cmd $noise > quit {$noise})
```

- [.] optional
- {.} zero or more
- (.) block
- <.> loop
- <<.>> context dep. loop
- .|. alternative

Grammar definition (HParse)

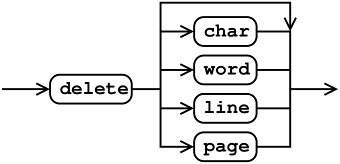
```
> cat grammar.bnf
$dir  = up | down | left | right;
$mcmd = move $dir | top | bottom;
$item = char | word | line | page;
$dcmd = delete [$item];
$icmd = insert;
$ecmd = end [insert];
$cmd  = $mcmd | $dcmd | $icmd | $ecmd;
$noise     = sil | fil | spk;
({$noise} < $cmd $noise > quit {$noise})
```



- [.] optional
- {.} zero or more
- (.) block
- <.> loop
- <<.>> context dep. loop
- .|. alternative

Grammar definition (HParse)

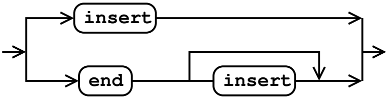
```
> cat grammar.bnf
$dir  = up | down | left | right;
$mcmd = move $dir | top | bottom;
$item = char | word | line | page;
$dcmd = delete [$item];
$icmd = insert;
$ecmd = end [insert];
$cmd  = $mcmd | $dcmd | $icmd | $ecmd;
$noise     = sil | fil | spk;
({$noise} < $cmd $noise > quit {$noise})
```



- [.] optional
- {.} zero or more
- (.) block
- <.> loop
- <<.>> context dep. loop
- .|. alternative

Grammar definition (HParse)

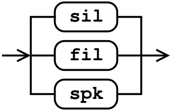
```
> cat grammar.bnf
$dir  = up | down | left | right;
$mcmd = move $dir | top | bottom;
$item = char | word | line | page;
$dcmd = delete [$item];
$icmd = insert;
$ecmd = end [insert];
$cmd  = $mcmd | $dcmd | $icmd | $ecmd;
$noise     = sil | fil | spk;
({$noise} < $cmd $noise > quit {$noise})
```



- [.] optional
- {.} zero or more
- (.) block
- <.> loop
- <<.>> context dep. loop
- .|. alternative

Grammar definition (HParse)

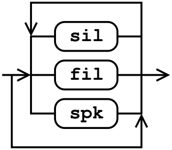
```
> cat grammar.bnf
$dir  = up | down | left | right;
$mcmd = move $dir | top | bottom;
$item = char | word | line | page;
$dcmd = delete [$item];
$icmd = insert;
$ecmd = end [insert];
$cmd  = $mcmd | $dcmd | $icmd | $ecmd;
$noise     = sil | fil | spk;
({$noise} < $cmd $noise > quit {$noise})
```



- [.] optional
- {.} zero or more
- (.) block
- <.> loop
- <<.>> context dep. loop
- .|. alternative

Grammar definition (HParse)

```
> cat grammar.bnf
$dir  = up | down | left | right;
$mcmd = move $dir | top | bottom;
$item = char | word | line | page;
$dcmd = delete [$item];
$icmd = insert;
$ecmd = end [insert];
$cmd  = $mcmd | $dcmd | $icmd | $ecmd;
$noise     = sil | fil | spk;
({$noise} < $cmd $noise > quit {$noise})
```



- [.] optional
- {.} zero or more
- (.) block
- <.> loop
- <<.>> context dep. loop
- .|. alternative

Grammar parsing (HParse) and recognition (HVite)

Parse grammar

```
> HParse grammar.bnf grammar.slf
```

Run recognition on file(s)

```
> HVite -C offline.cfg -H mono_32_2.mmf -w grammar.slf
        -y lab dict.txt phones.lis audio_file.wav
```

Run recognition live

```
> HVite -C live.cfg -H mono_32_2.mmf -w grammar.slf
        -y lab dict.txt phones.lis
```

Evaluation (HResults)

```
> HResults -I reference.mlf ... word.lst recognized.mlf
```

```
====================== HTK Results Analysis =======================
  Date: Thu Jan 18 16:17:53 2001
  Ref : nworkdir_train/testset.mlf
  Rec : nresults_train/mono_32_2/rec.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=74.07 [H=994, S=348, N=1342]
WORD: %Corr=94.69, Acc=94.37 [H=9202, D=196, S=320, I=31, N=9718]
===================================================================
```

N = total number, I = insertions, S = substitutions, D = deletions

correct: $H = N - S - D$

%correct: $\%Corr = H/N$ **accuracy**: $Acc = \frac{H-I}{N} = \frac{N-S-D-I}{N}$