# Pokročilé metody rozpoznávání řeči

Přednáška 3

Fonémové HMM a jejich trénování

rozpoznávání izolovaných slov

# Kontrola nahrávek

**TEXT: budeme žádat o dotaci která by mohla činit až dvacet dva milionů**
**ASR: budeme žádat o dotaci která by mohla činit až dvacet dva milionu**
**91.67( 91.67) [H= 11, D= 0, S= 1, I= 0, N= 12, (OOV= 0)]**
**ASTr: 2 budeme žádat o dotaci kterábi mohla činit až dvacet dva milijonu 2**

**TEXT: utrpěly i parky a lesy kde vítr způsobil pády větví i stromů**
**ASR: utrpěli i paniky ale si kde vítr způsobil pády větví stromů**
**58.33( 58.33) [H= 7, D= 1, S= 4, I= 0, N= 12, (OOV= 0)]**
**ASTr: 2 - utrpjeli i paniki ale si gde vítr spúsobil pádi vjetví stromú 2**

**TEXT: stíhání začalo v obvodu Poruba vozy se pak dostaly až do centra Ostravy**
**ASR: stíhání začalo v obvodu Poruba vozy se pak dostali až do centra Ostravy**
**92.31( 92.31) [H= 12, D= 0, S= 1, I= 0, N= 13, (OOV= 0)]**
**ASTr: 2 sťíháňí začalo f obvodu poruba vozi sepag dostali aš do centra ostravi 2**

**TEXT: a protože sedmička je šťastné číslo vypadá to s tím komiksem zatím náramně**
**ASR: a protože je Sedmička šťastné číslo vypadá to s tím komiksem zatím náramně**
**92.31( 84.62) [H= 12, D= 1, S= 0, I= 1, N= 13, (OOV= 0)]**
**ASTr: 2 - 0a protožeje sedmička šťastné číslo vipadá to sťím komiksem 2 zaťím náramňe 2**

**TEXT: pokud tedy někde vede cesta po frekventované silnici není to dost často naše vina**
**ASR: pokud tedy vede někde cesta po frekventované silnici není to dost často naše vina**
**92.86( 85.71) [H= 13, D= 1, S= 0, I= 1, N= 14, (OOV= 0)]**
**ASTr: 2 pokut tedi vede ňegde 2 cesta po frekventované silňici neňíto dost často naše vina 2**

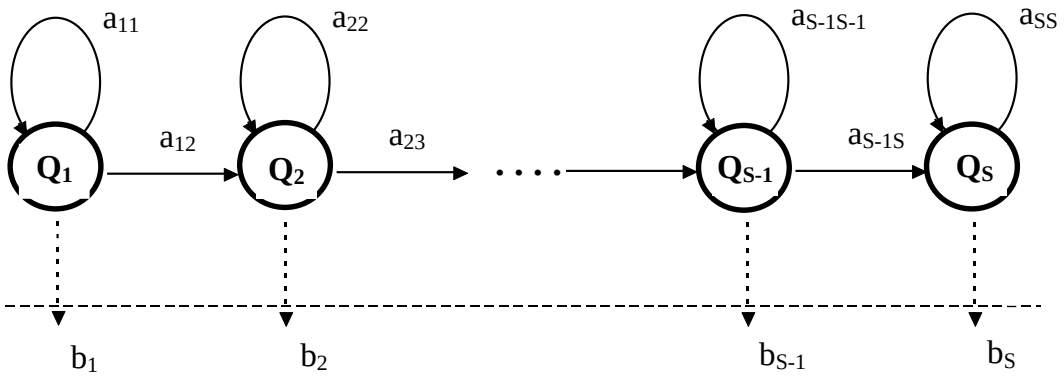
# Co byste měli mít připravené?

1. Každý 100 nahrávek. Nahrávky by měly obsahovat všechny fonémy (Zkontrolujte si.)
2. Ke každé nahrávce TXT (textová podoba věty) soubor a PHN (fonetická podoba věty). Pokud soubor PHN obsahuje i symboly hluků (0,1,..5), nahraďte je symbolem ticha (-).
3. Ke každé nahrávce soubor LAB (automatická konverze z PHN).
4. Nahrávky si mezi sebou nasdílejte.
5. Z minulého semestru testovací sety SD a SI (nahrávky číslovek 0 až 9)
   - SD: 50 vlastních nahrávek
   - SI: nahrávky osob 30-49
6. Na e-learningu připravena další data pro trénování: Data-PMR-old.zip
   15 osob x 100 nahrávek (WAV+TXT+PHN) – celkem cca 2 hodiny

# Skryté Markovovy modely (HMM)

**Metoda HMM (Hidden Markov Model – skryté Markovovy modely)** reprezentuje řeč (*slovo, hlásku, celou promluvu*) **stavovým modelem** s pravděpodobnostními parametry

Typická **struktura slovního HMM** – takzvaný ***levo-pravý model***




$Q_s$ ... **stavy** (šipky naznačují možné přechody mezi nimi)

$a_{ij}$ .... **přechodová pravděpodobnost** – pravděpodobnost, že (v aktuálním framu) model přejde ze stavu *i* do stavu *j*

$b_s(\mathbf{x})$...**výstupní pravděpodobnostní rozložení** – funkce určující pravděpodobnost, že příznakový vektor **x patří ke stavu s** *(nejčastěji gaussovské rozložení)*

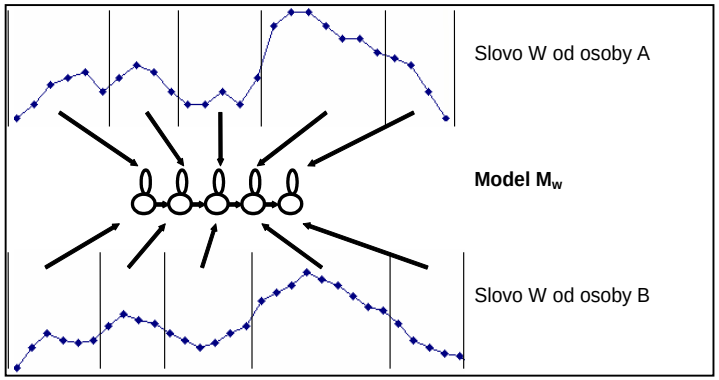
# Jak určit parametry výstupního rozložení?

*Pro zjednodušení výkladu uvažujme 1-rozměrný příznakový vektor* **x**

Předpokládáme, že $b_s(x)$ má **gauss. rozdělení**

$$b_s(x) = \frac{1}{\sqrt{2\pi}\cdot\sigma_s}\exp\left[-\frac{(x-\mu_s)^2}{2\sigma_s{}^2}\right]$$

Mějme **alespoň 2 nahrávky pro každé slovo**
(čím více nahrávek,
tím lepší model získáme)




**Pokud víme, které framy patří ke jednotlivým stavům**
(jak ukázáno na obrázku)

pak **stř. hodnotu** určíme jako $\mu_s = \frac{1}{N_s}\sum_{n=1}^{N_s} x_n$ a **rozptyl** jako $\sigma_s^2 = \frac{1}{N_s}\sum_{n=1}^{N_s}(x_n-\mu_s)^2$

kde $N_s$ je počet framů přiřazených stavu $s$

**Gaussovské rozložení pro vícepříznakový vektor x**

$$b_s(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^P \det\Sigma_s}}.\exp[-\tfrac{1}{2}(\mathbf{x}-\bar{\mathbf{x}}_s)^T\boldsymbol{\Sigma}_s^{-1}(\mathbf{x}-\bar{\mathbf{x}}_s)]$$

$$\boldsymbol{\mu}_s = \frac{1}{N_s}\sum_{n=1}^{N_s}\mathbf{x}_n$$

$$\boldsymbol{\Sigma}_s = \frac{1}{N_s}\sum_{n=1}^{N_s}(\mathbf{x}_n-\boldsymbol{\mu}_s)(\mathbf{x}_n-\boldsymbol{\mu}_s)^T$$

# Význam parametrů u vícepříznakových vektorů

$$\boldsymbol{\mu}_s = \frac{1}{N_s}\sum_{n=1}^{N_s}\mathbf{x}_n$$

**vektor středních hodnot (angl. Mean)**

$$\boldsymbol{\Sigma}_s = \frac{1}{N_s}\sum_{n=1}^{N_s}(\mathbf{x}_n - \boldsymbol{\mu}_s)(\mathbf{x}_n - \boldsymbol{\mu}_s)^T$$

**kovarianční matice (Covariance matrix)**

$$\boldsymbol{\Sigma}_s = \begin{pmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{1p} \\ \rho_{21} & \rho_{22} & & \rho_{2p} \\ \cdots & & & \\ \rho_{p1} & \rho_{p2} & & \rho_{pp} \end{pmatrix}$$

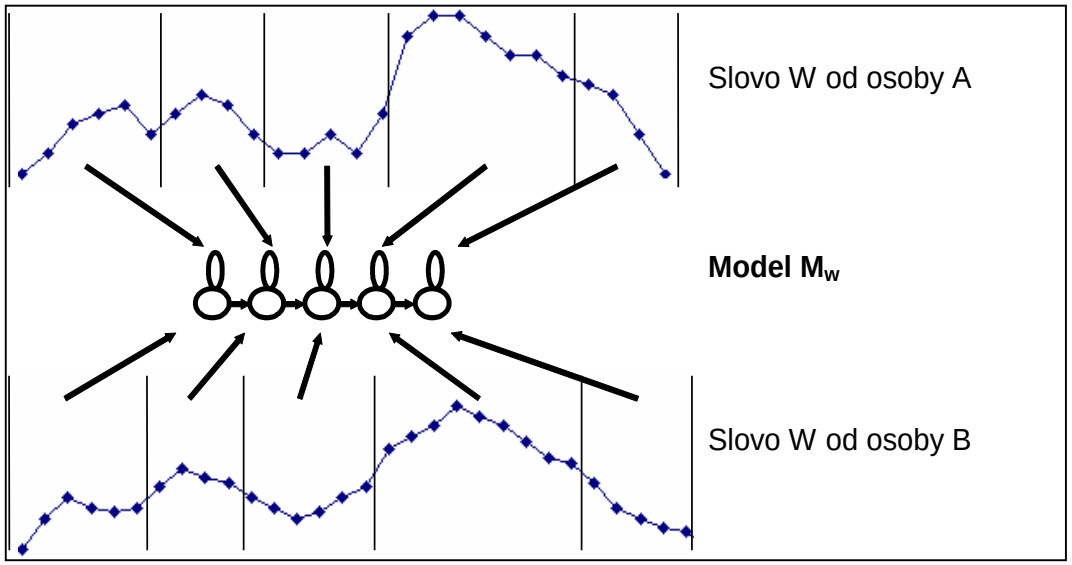
**na hlavní diagonále leží rozptyly příznaků,**

**na ostatních pozicích jsou kovariance („vzájemné rozptyly“)**

**pro nekorelované příznaky jsou hodnoty mimo diagonálu malé a lze je zanedbat**

**Kepstrální příznaky** se vyznačují malou vzájemnou korelovaností,
a proto **místo** kompletní **matice** lze použít pouze **vektor diagonálních hodnot**,
výpočet se pak významně usnadní a urychlí

# Jak určit přechodové pravděpodobnosti?




**Pravděpodobnost přechodu** ze stavu *s* do *s+1*

$$a_{ss+1} = \frac{K}{N_S}$$

kde *K* je počet výstupů ze stavu s
(je vlastně roven počtu nahrávek *K* daného slova)

**Pravděpodobnost setrvání** (v tomtéž stavu *s*)

$$a_{ss} = 1 - a_{ss+1}$$

# Jak skutečně trénovat parametry HMM?

**Ve skutečnosti nevíme** který frame patří k jakému stavu.
(Z tohoto důvodu se metodě HMM říká ***skryté*** markovské modely)

Metoda **trénování HMM** (tj určování jejich parametrů) je proto **iterativní**

1. **Inicializační krok**
   Framy všech nahrávek daného slova jsou rovnoměrně přiděleny jednotl. stavům,
   z nich pak určíme stř. hodnoty, rozptyly a přechodové pravděpodobnosti

2. **Přiřazovací krok**
   s využitím aktuálního modelu a Viterbiho algoritmu (popsán loni) nalezneme
   nové (už ne rovnoměrné ale obvykle lepší) přiřazení mezi framy a stavy

3. **Reestimační krok**
   pro toto nové přiřazení určíme stř. hod., rozptyly a přechod. pravděpodobnosti

4. **Opakování, případně konec**
   pokud se nové stř. hod., rozptyly a přech. pravd. liší od předchozích
   anebo pokud se celkové skóre modelu liší o více než ε od předešlého,
   anebo pokud je toto skóre horší než předešlé
   jdeme zpět na krok 2, jinak ukončíme trénování

# Jak ještě lépe trénovat parametry HMM?

## Baumův-Welchův (forward-backward) algoritmus

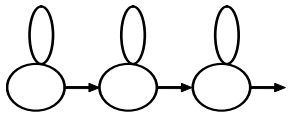
- framy nejsou *pevně a výlučně* přiřazeny k jednotlivým stavům,
- naopak, každý frame se s *jistou pravděpodobností* může podílet na parametrech všech stavů
- vztahy pro výpočet stř. hod., rozptylů a přechod. pravděpodobností nově obsahují ještě tzv. *okupační pravděpodobnosti*
- ty se dají určit na základě tzv. **dopředné** a **zpětné** pravděpodobnosti α a β přesné vztahy lze najít v HTKbook – kapitola 8.

## Praktický postup trénování HMM

1. Rovnoměrné rozdělení framů ke stavům, výpočet inicializačních hodnot parametrů
2. Iterační postup zpřesňování parametrů základním přístupem (2 – 10 iterací)
3. Iterační postup zpřesňování parametrů Baum-Welch algoritmem (2 – 10 iterací)

# Fonémové HMM

## Třístavová struktura modelu

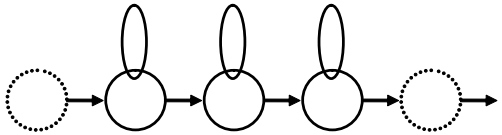


přibližně odpovídá situaci:

1. stav - přechod z předchozích fonémů
2. stav - jádro fonému
3. stav - přechod do dalších fonémů

( u monofonů mají 1. a 3. stavy velkou variabilitu, u trifonů pak mnohem menší

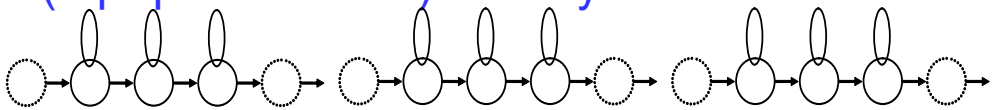
## Struktura modelu používaná v HTK



celkem 5 stavů

1. a 5. stav je fiktivní (vstupní a výstupní)
   slouží ke snazší implementaci přechodů mezi modely hlásek
2.-4. stav – význam jako výše

Slovní (a případně větné) modely složené z hláskových modelů



# Trénování fonémových modelů v HTK (1)

## 1. Vytvoření prototypu modelu

textový popis struktury modelu
s tagy a čísly (číselné hodnoty nehrají roli)

Prototyp se rozkopíruje do modelů všech hlásek
a sloučením se vytvoří jediný soubor modelů

hmmdefs

```
~h "aa"
   <BeginHMM> ...
   <EndHMM>
~h "eh"
   <BeginHMM> ...
   <EndHMM>
... etc
```

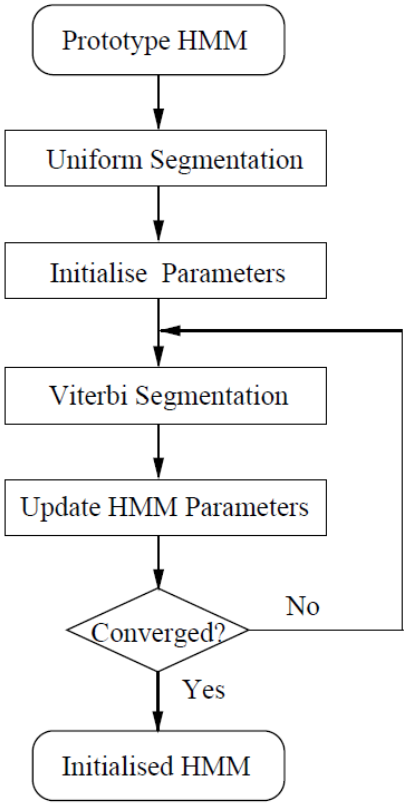
```
<BeginHMM>
 <NumStates> 5
 <State> 2
    <Mean> 39
      0.0 0.0 0.0 ...
    <Variance> 39
      1.0 1.0 1.0 ...
 <State> 3
    <Mean> 39
      0.0 0.0 0.0 ...
    <Variance> 39
      1.0 1.0 1.0 ...
 <State> 4
    <Mean> 39
      0.0 0.0 0.0 ...
    <Variance> 39
      1.0 1.0 1.0 ...
 <TransP> 5
  0.0 1.0 0.0 0.0 0.0
  0.0 0.6 0.4 0.0 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>
```

# Trénování fonémových modelů v HTK (2)

## 2. Je-li známé umístění hlásek v nahrávce

*(v souboru *.lab jsou přesně uvedeny začátky a konce)*

0000 3600 si
3600 4200 a
4200 4700 h
4700 5300 o
5300 5700 j
….




Použije se program **Hinit**

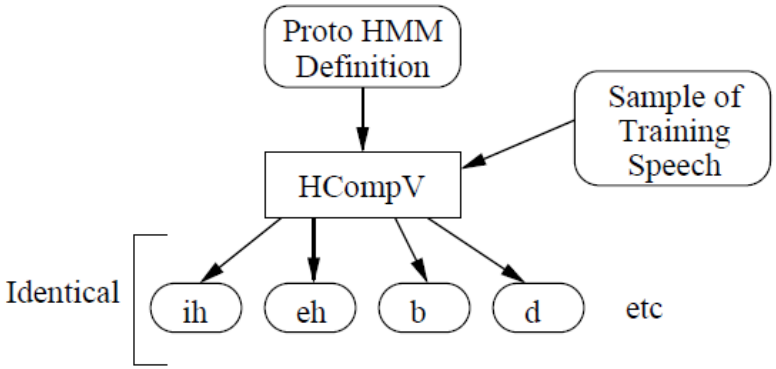
- ten „vyřízne“ všechny realizace každé hlásky a pro každou hlásku iterativně natrénuje parametry jejího modelu

# Trénování fonémových modelů v HTK (3)

## 3. Není-li známé umístění hlásek

*(v souboru *.lab nejsou uvedeny začátky a konce – resp. uvedeny fiktivní časy)*

0000 0000 si
0000 0000 a
0000 0000 h
0000 0000 o
0000 0000 j
….




Použije se program **HCompV**

- ten provede tzv. Flat Start („plochý start“)
přes všechny nahrávky určí hodnoty kovarianční matice (rozptyly) a umístí je do modelů všech hlásek

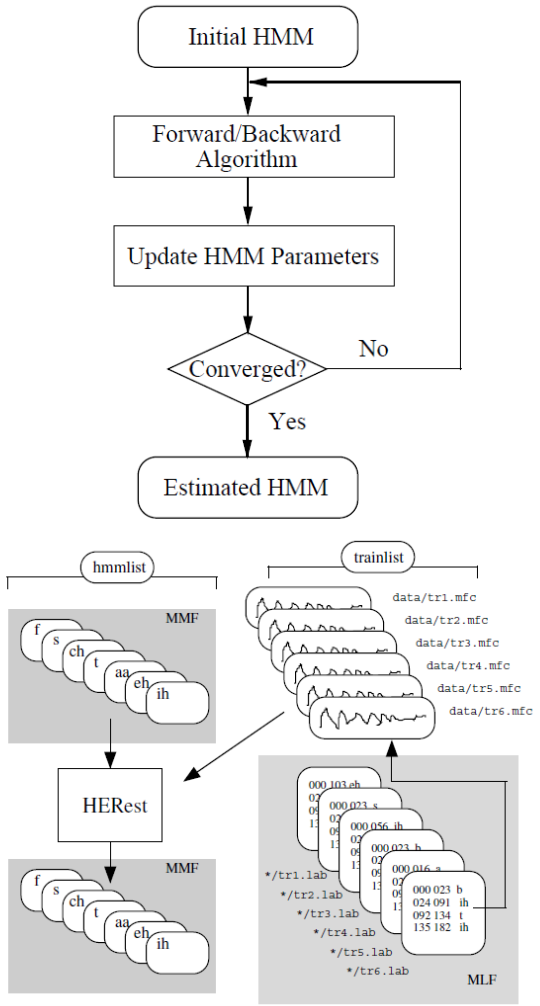
cílem je alespoň „nějak“ inicializovat hodnoty parametrů

# Trénování fonémových modelů v HTK (4)

## 4. Reestimace parametrů modelů

Použije se několik iterací programem **HERest**

- ten si na za základě informace v souboru *.lab sestaví model celé nahrávky zřetězením všech dílčích hláskových modelů
- pro každou nahrávku určí dílčí příspěvek k výpočtu parametrů pomocí B-W algoritmu,
- toto zopakuje se všemi nahrávkami
- na závěr každé iterace se spočítají hodnoty všech parametrů modelů
- toto se provede v několika iteracích za sebou



# Texty k nastudování

## HTKbook

Kapitola 3. A Tutorial Example of Using HTK

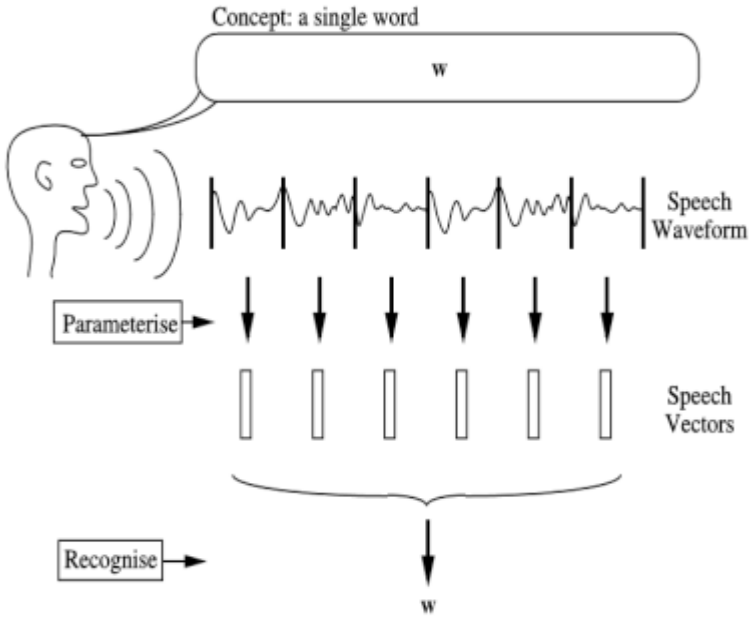
Kapitola 8. HMM Parameter Estimation

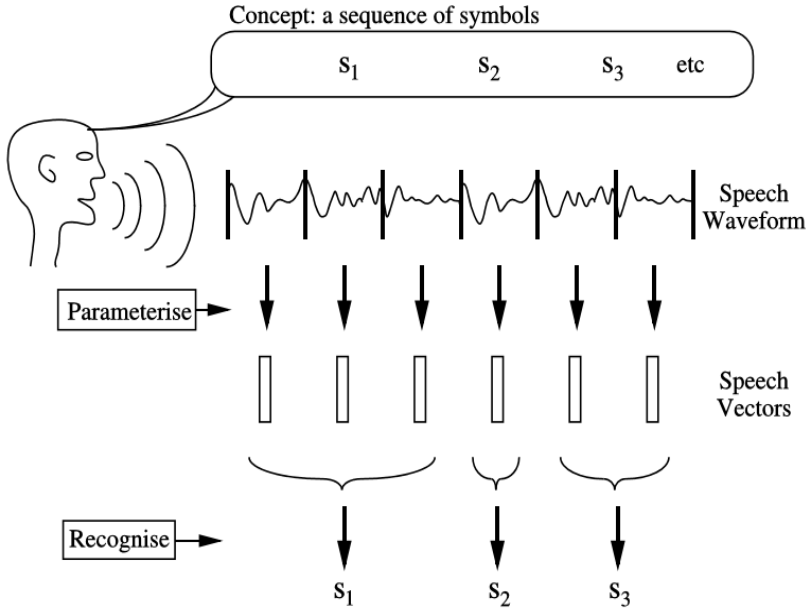
Podklady k předmětu PMR (trénování celoslov. modelů) – stačí modifikovat podle dnešní přednášky
na e-learningu: Podklady_pro_trenovani_PMR.zip

K dispozici na e-learningu: HTK-trenovani-skripty.zip
(skripty v Perlu + příklady některých souborů)

# HTK – princip rozpoznávání řeči

## Rozpoznávání izolovaných slov a spojité řeči (sekvence)




Prostředí HTK je zaměřeno na **rozpoznávání spojité řeči** (sekvence slov),
- izolovaná slova jsou brána jako speciální případ (jako sekvence ticho – slovo – ticho)

# HTK – slovník a slovní síť

## Slovník definuje seznam slov a z jakých (dílčích) jednotek se skládají

Příklady: slovník pro rozpoznávání číslic vytvořený z celoslovních a hláskových jednotek

| slovo | jednotka |
|---|---|
| NULA | nula |
| JEDNA | jedna |
| DVA | dva |
| …. | |
| DEVET | devet |
| SENT-END | [] sil |
| SENT-START | [] sil |

symbol pro ticho

Minulý semestr

| slovo | jednotky |
|---|---|
| NULA | n u l a |
| JEDNA | j e d n a |
| DVA | d v a |
| …. | |
| DEVET | d e v j e t |
| SENT-END | [] sil |
| SENT-START | [] sil |

## Gramatika – symbolický popis povolených sekvencí slov

$digit = JEDNA | DVA | TRI | CTYRI | PET | SEST | SEDM | OSM | DEVET | NULA;

( SENT-START ( $digit ) SENT-END ) promluva musí obsahovat právě 1 číslici

---------- alternativně----------

( SENT-START ( <$digit> ) SENT-END ) promluva může obsahovat 1 nebo více číslic

## Slovní síť – interní popis mezislovních přechodů

HParse grammar wordnet

# HTK – rozpoznávání

## Pro rozpoznávání se použije program **HVite**

## Příklad volání:

HVite -H hmm6/hmmdefs -S test.scp -i recout.mlf -w wordnet -p -70.0 -s 0 dict models0

## Ve výše uvedeném příkladu je

hmmdefs … soubor obsahující všechny natrénované modely v 6. iteraci

test.scp …. seznam zparametrizovaných testovacích nahrávek

dict … slovník,  wordnet …síť, models0 … seznam použitých modelů

## Výstup je v souboru recout.mlf a vypadá následovně

"D:/HTK/DATA/0000_MVL/c0_p0000_s04.rec"
0 6700000 SENT-START -4.080025
6700000 11900000 NULA -877.901184
11900000 19800000 SENT-END 17.283134
.
"D:/HTK/DATA/0000_MVL/c1_p0000_s04.rec"
0 5900000 SENT-START -8.679775
5900000 13200000 JEDNA -1429.232910
13200000 19800000 SENT-END 22.320038

# HTK – vyhodnocování experimentů

## Pro vyhodnocování se použije program **HResult**

## Příklad volání:

HResults -e ??? SENT-START -e ??? SENT-END -t -I testref.mlf models0 recout.mlf

## Ve výše uvedeném příkladu je

recout.mlf … výstup rozpoznávače, testref.mlf … soubor obsahující slova v každé nahrávce

## Výstup vypadá následovně

```
Aligned transcription: D:/HTK/DATA/0000_MVL/c2_p0000_s04.lab vs D:/HTK/DATA/0000_MVL/c2_p0000_s04.rec
 LAB: DVA
 REC: NULA
Aligned transcription: D:/HTK/DATA/0000_MVL/c5_p0000_s04.lab vs D:/HTK/DATA/0000_MVL/c5_p0000_s04.rec
 LAB: PET
 REC: DEVET
….
====================== HTK Results Analysis =======================
  Date: Fri Mar 29 14:31:16 2019
  Ref : testref.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=90.00 [H=45, S=5, N=50]
WORD: %Corr=90.00, Acc=90.00 [H=45, D=0, S=5, I=0, N=50]
================================================================
```

# Úloha do příště

## Rozpoznávač číslic založený na HTK

1) Vytvořte si potřebné soubory a skripty (nebo dávky) pro trénování fonémových modelů a následné rozpoznávání slov.
2) Na **vašich vlastních datech** (zparametrizovaných pomocí HCopy) si natrénujte fonémové modely (všechny hlásky a ticho)
3) Na nahrávkách číslic z minulého semestru proveďte testy rozpoznávání (SD s vlastními daty a SI s daty ostatních mluvčích)
4) Nyní **rozšiřte trénovací sadu** o data sdílená mezi vámi a o data z e-learningu (15 x 100 vět) a natrénujte opět fonémové modely.
5) S těmito modely proveďte podobný experiment jako ad 3)
6) Do konce neděle mi zašlete výsledky.