Pokročilé metody rozpoznávání řeči

Přednáška 5

Rozpoznávání sekvencí slov pomocí HMM a HTK

Výsledky a zkušenosti z předchozí úlohy

Výsledky jednoho z vás

Statistiky trénovací množiny dat.

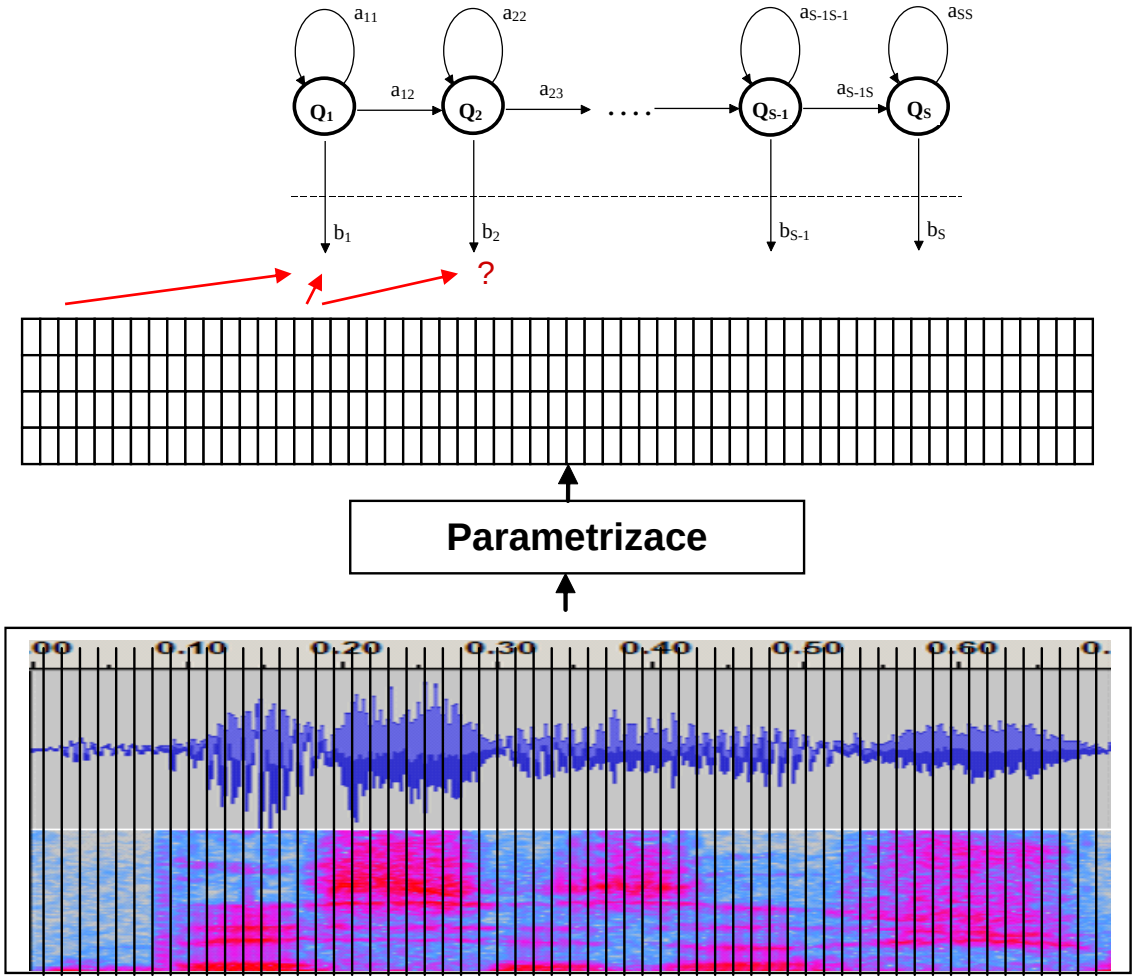
Pocet osob	95
Pocet wav	9514
Tren. hodin	15,96

	SD [%]	SI [%]	Jmena [%]	n-iter
mono 1	10	10	4,17	0
mono 2	100	87,1	91,15	8
mono 4	100	91,6	94,79	8
mono 8	100	92,4	94,27	8
mono 16	100	91,6	97,4	8
mono 32	100	91,6	96,88	8

Poznatky z trénování (počty mixtur, iterací, doba …)?
Doba testování?
Tvorba nového slovníku?

Připomenutí principů IWR pomocí HMM(1)

Cílem je určit, s jakou pravděpodobností by modely slov ve slovníku vygenerovaly sekvenci příznakových vektorů rozpoznávaného (neznámého) slova



Model

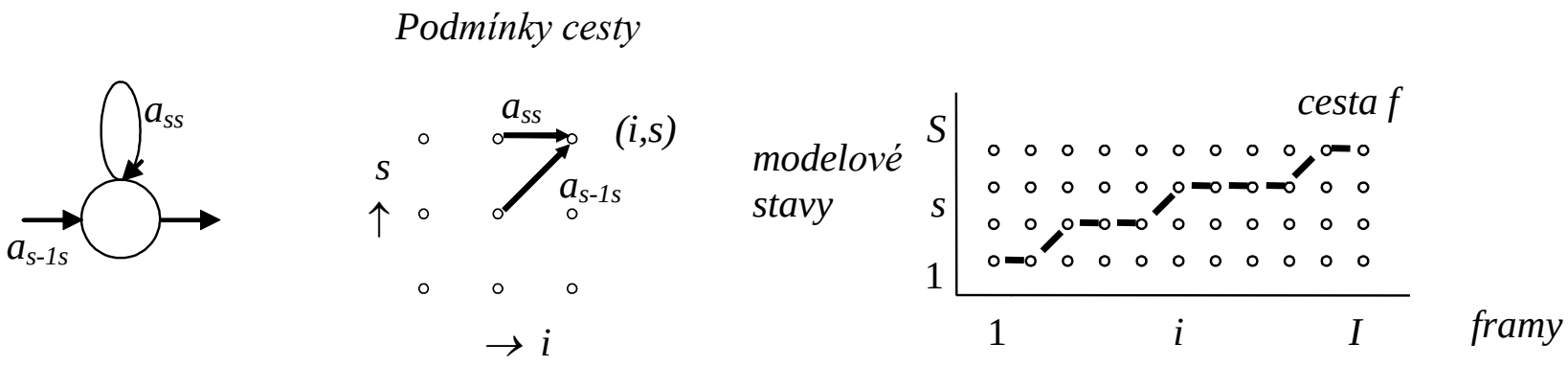
Příznakové vektory

Rozpoznávané slovo

Připomenutí principů IWR pomocí HMM(2)

Výpočet je podobný jako u metody DTW

Hledáme **nejlepší cestu** (přiřazovací funkci ***f*** *s nejvyšší pravděpodobností*), tentokrát ovšem ***v rovině signálových framů a modelových stavů***



Pravděpodobnostní „skóre“ pro výše uvedenou cestu *f* určíme jako

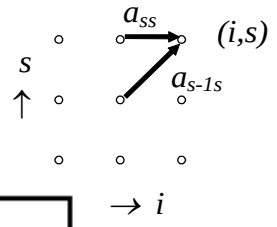
$$P(\mathbf{X},\mathbf{M}) = b_1(x_1)\cdot a_{11}b_1(x_2)\cdot a_{12}b_2(x_3)\cdot a_{22}b_2(x_4)....$$

Připomenutí - Viterbiho algoritmus

Jde o základní algoritmus rozpoznávání i trénování HMM

Definujme - kumulované skóre v bodě (i, s):

$$V(i,s) = b_s(x_i) \cdot Max[a_{ss}V(i-1,s), a_{s-1s}V(i-1,s-1)]$$



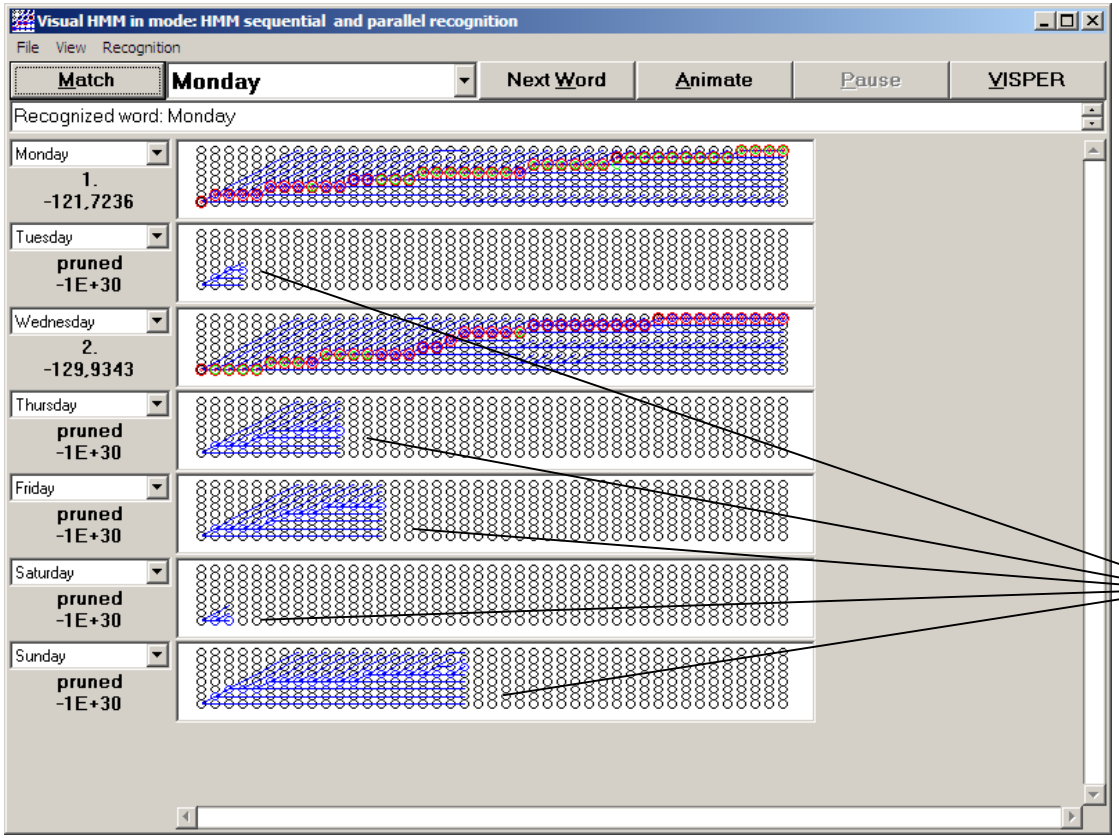
Viterbi algorithm	
Step 1:	*Initialization*
	$V(1,1) = b_1(x_1) \quad V(1,s) = -\infty$ pro $s=2,....S \quad B(1,1) = 1$
Step 2:	*Recursion*
	For $i = 2, ... I$
	For $s = 1, ... S$
	$O(k) = a_{ks}V(i-1,k)$ pro $k = s-1, \ s$ *(temporary variable)*
	$V(i,s) = b_s(x_i) \cdot \underset{k}{Max}[O(k)] \quad B(i,j) = \underset{k}{ArgMax}[O(k)]$
Step 3:	*Termination*
	$P(\mathbf{X},\mathbf{M}) = V(I,S)$
Step 4:	*Backtracking*
	$f(I) = S$ for i $= I-1,....1 \quad f(i) = B(i+1, f(i+1))$

Část „Backtracking“ není nutná pro rozpoznávání (izolovaných) slov, ale je klíčová pro trénování a pro rozpoznávání sekvencí slov

Paralelní implementace & prořezávání

Klasifikace nemusí nutně probíhat **sekvenčně** (model po modelu), může být implementována i **paralelně** (všechny modely současně, frame po framu).

Princip: Po zpracování každého framu najdeme **stav s nejvyšším kumul. skóre** a dále modely, jejichž nejlepší kum. **skóre je výrazně horší.** S nimi přestaneme dále počítat, protože už **nejsou kompetitivní.** Technika se nazývá **prořezávání (pruning)**.



Nastavením **vhodného prořezávacího prahu (threshold)** rozpoznávání se stane **rychlejší, aniž** by došlo ke **zhoršení výsledků.**

Výpočet u těchto modelů skončil předčasně díky prořezávání.

Rozdíl mezi IWR a CWR

CWR (Connected Word Recognition)

U CWR neplatí okrajové podmínky IWR

1. slovo nemusí začínat v prvním a končit v posledním framu, naopak může začínat i končit kdykoliv,
2. slova mohou být za sebou řazena úplně libovolně nebo podle určitých pravidel (např. gramatika, nebo pravděpodobnostní jazykový model),
3. v nahrávce se mohou objevovat i neřečové události (ticho, hluk)

Lze využít IWR rozpoznávač i pro CWR?

- Malý slovník (např. číslice) a pevně dané pořadí nebo počet slov (např. 3)

- V takovém případě můžeme složit modely všech kombinací slov (v případě číslovek 10^3 modelů) a provést s nimi klasické rozpoznávání (např. paralelním způsobem). Výsledkem bude sekvence s největší pravděpodobností.

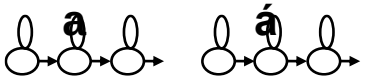
- Je to samozřejmě neefektivní a v praxi málokdy použitelné.

Rozpoznávání sekvence slov a spojité řeči
s využitím gramatiky nebo jazykového modelu

Fonetický inventář
a, á, b, c, č, X, …… z, ž,

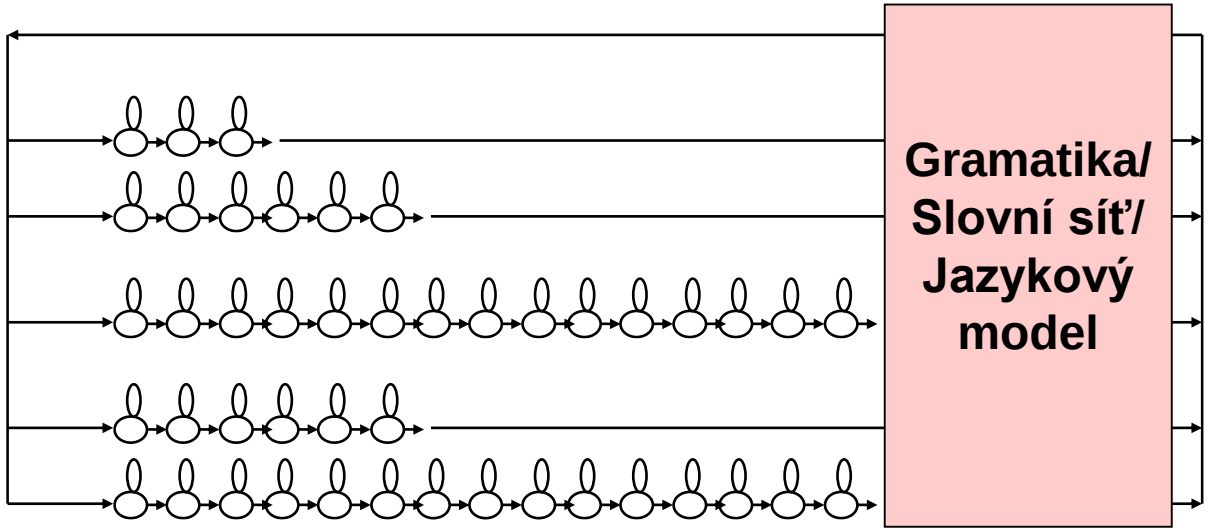
Inventář hluků
ticho, nádech, klik, …

Akustické modely



Lexikon
“ticho”
ať
…..
robot
…
už
…..
Zürich

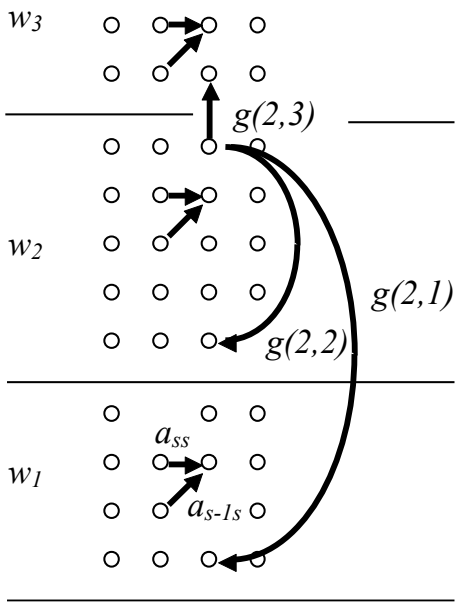
Výslovnost
-
ať
…..
robot
…..
uš
ciriX

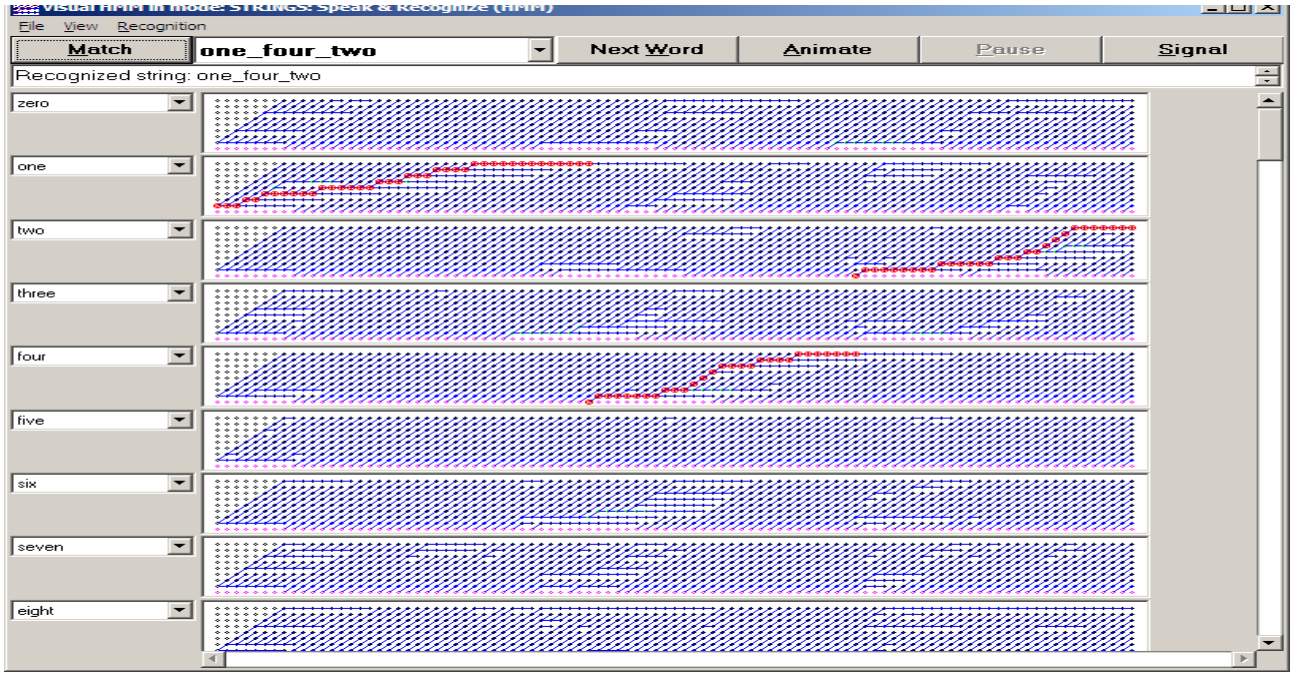


*Pozn.: Síť všech modelů slov je nyní “zacyklena”, tj. z posledního stavu každého slova vede cesta na počáteční stavy všech slov. Možnost či nemožnost přechodu mezi slovy, případně jeho pravděpodobnost, udává **jazykový model**.*

Viterbiho dekodér pro sekvenci slov

Rozpoznávání se děje pomocí **Viterbiho algoritmu**, který běží **paralelně pro všechna slova**. **Uvnitř slova** se provádí klasický výběr ze dvou předchozích stavů. V každém framu **nejvyšší pravděpodobnost** z posledních stavů slov je přenesena do počátečních stavů všech slov. Nejlepší sekvenci slov najdeme zpětným trasováním po dosažení konce.





Pro každý frame a stav musíme v paměti udržovat informaci nejen o kumulovaném skóre, ale také čas (tj. číslo framu), kde daná instance slova začala.

Rozpoznávání (CWR) v HTK

Prakticky stejné jako bylo u IWR - použijeme program HVite

V případě, že řeč se v dané aplikaci řídí jednoduchou gramatikou, popíšeme gramatiku odpovídajícími příkazy a následně z ní vygenerujeme slovní síť - wdnet.txt

Vygenerování sítě již znáte z úlohy rozpoznávání izolovaných číslovek – více se můžete dozvědět v HTK book.

- Kapitola 3. A Tutorial Example of Using HTK
- Kapitola 11. Networks, Dictionaries and Language Models

Nastudujte si význam parametrů –s a –p v příkazu Hvite. U rozpoznávání sekvencí slov hrají důležitou roli.

Vyhodnocování rozpoznávání řeči (1)

V případě rozpoznávání izolovaných slov je vyhodnocování jednoduché:

$$Correctness = \frac{H}{N}.100\,\%$$

H …. počet správně rozp. slov, N …. počet všech testovaných slov

V případě rozpoznávání sekvencí slov je vyhodnocování složitější:

Příklad:

řečeno:	Zastavím	se	u	vás	zítra	dopoledne	(6 slov)
rozpoznáno	Za stavím	se		vás	zítra	do poledne	(7 slov)
	I	S	H	D	H	H	I	S	

H …. (hit) správně rozpoznáno, S …. (substituce) nesprávně rozpoznáno

D … (delece) vynecháno, I ….. (inzerce) vloženo

Vyhodnocování rozpoznávání řeči (2)

Při rozpoznávání sekvencí slov se používají dvě míry:

$$Correctness = \frac{H}{N}.100\,\%$$

$$Accuracy = \frac{H - I}{N}.100\,\%$$

Druhá míra je objektivnější, přísnější a za určitých podmínek může být < 0.

Jak zjistit hodnoty H, S, D, I?

Metoda minimální editační chyby (MEE)

- obdoba DTW
- hledá se cesta s nejmenším počtem oprav
- Levenshteinova vzdálenost
- před aplikací metody MEE je třeba stanovit „cenu“ chyb typu S, D, I (v HTK 10 – 7 – 7)

		k	i	t	t	e	n
	0	1	2	3	4	5	6
s	1	1	2	3	4	5	6
i	2	2	1	2	3	4	5
t	3	3	2	1	2	3	4
t	4	4	3	2	1	2	3
i	5	5	4	3	2	2	3
n	6	6	5	4	3	3	2
g	7	7	6	5	4	4	3

		S	a	t	u	r	d	a	y
	0	1	2	3	4	5	6	7	8
S	1	0	1	2	3	4	5	6	7
u	2	1	1	2	2	3	4	5	6
n	3	2	2	2	3	3	4	5	6
d	4	3	3	3	3	4	3	4	5
a	5	4	3	4	4	4	4	3	4
y	6	5	4	4	5	5	5	4	3

Vyhodnocování v HTK (1)

Program HResult:

Vezme výsledky rozpoznávače (nahrávku po nahrávce) a porovná je s referenčními (požadovanými) přepisy.

Příklad výpisu:

```
====================== HTK Results Analysis =======================
  Date: Sat Sep  2 14:14:22 1995
  Ref : refs
  Rec : results
---------------------- Overall Results ----------------------------
SENT: %Correct=98.50 [H=197, S=3, N=200]
WORD: %Corr=99.77, Acc=99.65 [H=853, D=1, S=1, I=1, N=855]
====================================================================
```

Uvádí počet správně rozpoznaných vět (věta = sekvence slov v jednom souboru) a počet správně rozpoznaných slov (Correctness a Accuracy)

Vyhodnocování v HTK (2)

Použití programu HResult:

1) Nahrají se testovací nahrávky a ke každé se připraví soubor s referenčním přepisem (implicitně *.lab)
2) Vytvoří se jeden velký referenční soubor přes všechny nahrávky ve formátu MLF (refer.mlf)
3) Provede se rozpoznávání všech testovacích nahrávek pomocí HVite (pomocí –S se specifikuje seznam souborů)
4) Výstupní soubor z rozpoznávání (result.mlf) se porovná s referenčním souborem

```
#!MLF!#
„test1.lab"
NULA
SEST
PET
.
„test2.lab"
PET
TRI
DEVET
.
```

HResults -C config -I refer.mlf monophones result.mlf

Samostatná úloha

Provést experimenty s rozpoznáváním sekvencí číslovek a sekvencí příkazů s použitím existujícího akustického modelu a jednoduché gramatiky.

Návod k řešení

Úloha 1:

1. Stáhněte si e-learningu testovací sadu obsahující nahrávky sekvencí číslic 0-9. (Celkem 90 nahrávek). Pocházejí od minulých studentů a měly by zahrnovat i soubory TXT LAB.
2. Dále si sestavte soubor refer.mlf
3. Proveďte rozpoznávací test s nejlepším modelem z minulého týdne a s odpovídající gramatikou a vyhodnoťte ho pomocí HRresult.
4. Experimentálně najděte vhodné hodnoty pro parametry -s a –p, abyste dosáhli nejlepšího skóre (největší hodnota Accuracy)

Úloha 2 (Spojovatelka):

1. Každý z vás si nahraje 20 nahrávek, v nichž budou věty typu: „Spojte mi (JP)“ nebo „Chtěl bych (JP)“ nebo „Prosím (JP)“ nebo „Dejte mi (JP)“. JP je jméno a příjmení (v příslušném pádu) – viz další slajd.
2. Sestavte si odpovídající slovník + gramatiku a opět proveďte test. (Gramatika je velmi podobná jako v tutorielu HTKBook).

Zaslání řešení

A) Do konce týdne mi pošlete řešení obou úloh a vaše data

B) Vaše data (WAV, TXT, LAB) dejte do adresáře Spojovatelka a zazipujte.

Jméno	**Příjmení**
Božena	Černá
Eva	Dvořáková
Adam	Novák
Petr	Svoboda
Jiří	Malý
Věra	Pokorná