HTK Trénování celoslovních modelů – úloha rozpoznávání číslic

1. **Vytvoření gramatiky**

Gramatika – zde míněna platná sekvence slov

Vytvoříme soubor ‘grammar‘ (pro případ, že v každé nahrávce je právě 1 vyslovená číslice)

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

( SENT-START ( $digit ) SENT-END )

Automaticky necháme vytvořit slovní síť:

Dávka **01\_MakeWordnet.bat**

obsahuje příkaz

HParse grammar wordnet

1. **Vytvoření slovníku**

Vytvoříme slovník – soubor ‘lexicon‘

CTYRI ctyri

DEVET devet

DVA dva

JEDNA jedna

NULA nula

OSM osm

PET pet

SEDM sedm

SENT-END [] sil

SENT-START [] sil

SEST sest

TRI tri

První sloupec slovníku obsahuje seznam slov. V seznamu musí vždy být 2 položky SENT-START a SENT-END, což jsou pomocné symboly pro rozpoznávač, označující začátek a konec promluvy. Je-li za řetězcem v prvním sloupci řetězec uzavřený v hranatých závorkách, vypíše rozpoznávač text v závorkách, jinak vypíše text v prvním sloupci. V uvedeném příkladu slouží hranaté závorky k tomu, aby na začátku a konci promluvy rozpoznávač nevypsal nic.

Druhý sloupec říká, jakým modelem bude slovo reprezentováno. V našem případě bude každé slovo reprezentováno svým jedinečným modelem, tedy např. slovo JEDNA bude reprezentováno modelem s názvem ‘jedna’.

Předpokládáme, že každá promluva má na začátku a konci ticho (šum pozadí). To bude mít také svůj model, zde ho označujeme ‘sil’ jako (silence).

Důležité:

1. **Názvy modelů musí obsahovat pouze znaky ASCII**
2. **Slovník musí být setříděn podle abecedy**

Dále musíme vytvořit soubor ‘wlist’, který obsahuje všechna slova ve slovníku ve stejném pořadí.

CTYRI

DEVET

….

TRI

Dále potřebujeme soubor 'global.ded', s následujícími příkazy:

MP sil sp sil

Připravíme dávku **02\_MakeDictionary.bat**

obsahující příkaz

HDMan -m -w wlist -n models0 -l dlog dict lexicon

Dávka vytvoří interní slovník nazvaný 'dict' a soubor models0, který je seznamem všech modelů, které se budou trénovat pro danou úlohu.

1. **Nahrávky a jejich popis**

Pro trénování modelů (a následně i pro jejich testování) musíme vytvořit nahrávky a k nim doprovodné soubory.

Parametry nahrávání 16 kHz, 16 bitů, mono.

Pro úlohu rozpoznávání pomocí celoslovních modelů se většinou vytvářejí nahrávky, obsahující vždy jedno slovo ze slovníku. Je třeba nahrát co nejvíce nahrávek od většího počtu osob.

Nahrávky mají např. název: XXX-01.wav, XXX-02.wav

Ke každé nahrávce musíme vytvořit ještě 2 soubory:

Textový soubor obsahující název slova, které bylo vyřčeno, tedy např. soubor nazvaný XXX-01.txt obsahuje jediný řádek a na něm slovo

JEDNA

Popisný soubor s obsahem toho, co je v nahrávce, tedy např. v předchozí nahrávce bylo řečeno slovo jedna, přičemž na začátku a na konci bylo ticho,

Soubor musí mít příponu .lab (od anglického label), takže např. XXX-01.lab obsahuje 3 řádky a na každém jméno elementu, který modelujeme, tedy

sil

jedna

sil

Trénovací set:

Z nahraných dat vyčleníme trénovací set a zbytek pak bude testovací set.

Soubor train.list obsahuje seznam trénovačích nahrávek, tj. souborů wav.

Důležité: HTK vyžaduje UNIXovou konvenci pro adresáře, tedy lomítko /

Dále vytvoříme soubor typu MLF (Master Label File), v němž budou uvedeny všechny trénovací soubory s příponou .lab a jejich obsah, ve formátu:

#!MLF!#

"cesta/XXX-01.lab"

sil

jedna

sil

.

"cesta/XXX-02.lab"

sil

dva

sil

.

Soubor nazveme 'source.mlf'

1. **Pametrizace nahrávek**

Musíme se rozhodnout pro konkrétní typ příznaků a ty pak musíme nechat spočítat pro každý soubor.

HTK nabízí několik typů příznaků, např.

FBANK – hodnoty log. energií ve zvoleném počtu pásem

MFCC – hodnoty kepstrálních koeficientů

u nichž je dále možné specifikovat, zda chceme též dynamické příznaky

např. MFCC\_0\_D\_A znamená, že se přidá též 0. koeficient, a dále Delta (první derivace) a A (druhá derivace)

Vytvoříme konfigurační soubor, v němž jsou hlavní údaje parametrizace:

Např. níže uvedený soubor **ParamConfig-FBANK** spočítá pro každý frame 16 koeficientů, které odpovídají energiím v 16 pásmech

# Coding parameters

SOURCEFORMAT = WAV

TARGETKIND = FBANK

TARGETRATE = 100000.0

SAVECOMPRESSED = F

SAVEWITHCRC = F

WINDOWSIZE = 250000.0

USEHAMMING = T

PREEMCOEF = 0.97

NUMCHANS = 16

ENORMALISE = F

Dále vytvoříme soubor 'param.list', v němž budou na každém řádku uvedeny dvojice souborů, první je vstupní soubor parametrizace, druhý výstupní soubor parametrizace (stejný název ale vhodně zvolená přípona), tedy např.

cesta/XXX-01.wav cesta/XXX-01.fbank

cesta/XXX-02.wav cesta/XXX-02.fbank

Vytvoříme dávku, v níž je parametrizační program HCopy aplikován na seznam param.list s nastavenou konfigurací ParamConfig-FBANK

**03\_DoParametrization.bat**

obsahuje příkaz

HCopy -T 1 -C ParamConfig-FBANK -S param.list

Pokud vše proběhlo správně, vytvořil se ke každé nahrávce soubor s příponou .fbank. Parametrizaci provedeme pro všechny soubory, trénovací i testovací.

1. **Trénování modelů**

Trénovací seznam:

Vytvoříme soubor ‘train.scp‘, v němž bude seznam všech zparametrizovaných souborů pro trénování, tedy např.

cesta/XXX-01.fbank

cesta/XXX-02.fbank

…

Nejprve musíme navrhnout strukturu modelů. Budou to levo-pravé modely s vhodně zvoleným počtem stavů.

U celoslovních modelů bývá počet stavů v rozmezí 4 – 20. U modelu ticha (sil) stačí 3 stavy. HTK k těmto skutečným stavům přidává ještě fiktivní vstupní a výstupní stav, což se zohlední v tzv. prototypovém souboru.

Prototypový soubor vypadá jako jakási šablona takto:

~o <VecSize> 16 <FBANK>

~h "proto"

<BeginHMM>

<NumStates> 5

<State> 2

<Mean> 16

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

<Variance> 16

1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0

<State> 3

<Mean> 16

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

<Variance> 16

1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0

<State> 4

<Mean> 16

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

<Variance> 16

1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 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>

Do šablony se zapíše počet stavů (skutečný počet+2)a dále počet a typ příznaků. U skutečných stavů (tedy kromě vstupního a výstupního) se u střed. hodnot (Mean) zapíše tolikrát 0.0, kolik je příznaků, a rozptylů zase tolikrát hodnota 1.0.

U matice přechodů se nenulovými hodnotami naznačí, které přechody jsou povoleny. Volí se zde odhadované hodnoty.

Připravil jsem prototyp 8-stavového modelu: proto-8s-16f

a 3-stavového modelu: proto-3s-16f

Oba jsou připraveny pro 16 příznaků

Připravíme si adresáře hmm0 až hmm6.

Zavoláme dávku 04\_ComputeVariance.bat

obsahuje příkazy

HCompV -C TrainConfig-FBANK -f 0.01 -m -S train.scp -M hmm0 proto-8s-16f

HCompV -C TrainConfig-FBANK -f 0.01 -m -S train.scp -M hmm0 proto-3s-16f

Soubor **TrainConfig-FBANK** je prakticky stejný jako **ParamConfig-FBANK**, pouze chybí 1. řádek.

Program HCompV vzal všechna trénovací data a určil na nich globální střední hodnotu a globální varianci. Tyto hodnoty budou použity jako inicializační při iteračním trénování a byly zapsány do obou prototypových souborů v adresáři hmm0.

V adresáři hmm0 vytvoříme prázdný soubor ‘hmmdefs‘. Do něj nakopírujeme obsah souboru proto-8s-16f tolikrát, kolik je modelů ve slovníku, a na řádek, kde se nachází ~h "proto…", místo slova "proto…" napíšeme název modelu, tedy např. "nula", "jedna". Na závěr nakopírujeme obsah souboru proto-3s-16f, místo slova "proto…" napíšeme název modelu "sil".

Soubor hmmdefs v adresáři hmm0 nyní obsahuje inicializované modely všech slov a ticha.

Nyní můžeme začít trénovat

Připravil jsem dávku **05\_TrainModels.bat**

která obsahuje řádky

HERest -C TrainConfig-FBANK -I train.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm0/hmmdefs -M hmm1 models0

HERest -C TrainConfig-FBANK -I train.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm1/hmmdefs -M hmm2 models0

HERest -C TrainConfig-FBANK -I train.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm2/hmmdefs -M hmm3 models0

HERest -C TrainConfig-FBANK -I train.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm3/hmmdefs -M hmm4 models0

HERest -C TrainConfig-FBANK -I train.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm4/hmmdefs -M hmm5 models0

HERest -C TrainConfig-FBANK -I train.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm5/hmmdefs -M hmm6 models0

Celkem 6x se v ní volá trénovací program HERest, který se řídí konfiguračním souborem TrainConfig-FBANK, a pro všechny soubory v trénovacím seznamu train.scp natrénuje všechny modely definované v seznamu models0. Při trénování se využívá soubor train.mlf, v němž je přesný popis jaké modely jsou obsaženy v jednotlivých souborech. Výsledkem každé iterace je soubor hmmdefs, který se zapíše vždy do následujícího adresáře. Poslední iterace je tedy v adresáři hmm6. (Adresář je třeba připravit předem.)

V tuto chvíli jsou všechny modely natrénovány.

1. **Testování rozpoznávače**

Vytvoříme si seznam zparametrizovaných testovacích nahrávek ‘test.scp‘.

Dávka **10\_RunTest.bat**

Obsahuje příkaz

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

HVite je program pro rozpoznávání (Viterbiho dekodér). Použije natrénované modely ze souboru hmmdefs, síť wordnet, seznam modelů models0, a na základě toho rozpozná všechny soubory v seznamu test.scp

Výsledek je v souboru recout.mlf, který vypadá nějak takto:

#!MLF!#

"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

.

Nyní zbývá vyhodnotit experiment. Spočívá to v porovnání toho, co bylo rozpoznáno, s tím, co mělo být rozpoznáno. Proto musíme vytvořit soubor ‘testref.mlf‘, který je hodně podobný trénovacímu souboru source.mlf. Obsahuje seznam testovacích souborů s uvedením toho, co v nich mělo být rozpoznáno.

#!MLF!#

"D:/HTK/DATA/0000\_MVL/c0\_p0000\_s04.lab"

NULA

.

"D:/HTK/DATA/0000\_MVL/c1\_p0000\_s04.lab"

JEDNA

.

Porovnání provede dávka 11\_ComputeResults.bat

která obsahuje řádek

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

Zavolá program HResults, který porovná soubory testref.mlf a recout.mlf a spočítá skóre. Přepínač –e říká, které položky se při vyhodnocování nemají brát v úvahu.

V dané konfiguraci program nejen spočítá skóre, ale vypíše též ty soubory, které nebyly rozpoznány správně.

D:\HTK\Pokus3>HResults -e ??? SENT-START -e ??? SENT-END -t -I testref.mlf models1 recout.mlf

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

Aligned transcription: D:/HTK/DATA/0002\_MJD/c0\_p0002\_s04.lab vs D:/HTK/DATA/0002\_MJD/c0\_p0002\_s04.rec

LAB: NULA

REC: DVA

Aligned transcription: D:/HTK/DATA/0002\_MJD/c1\_p0002\_s04.lab vs D:/HTK/DATA/0002\_MJD/c1\_p0002\_s04.rec

LAB: JEDNA

REC: DVA

Aligned transcription: D:/HTK/DATA/0003\_ZJL/c0\_p0003\_s04.lab vs D:/HTK/DATA/0003\_ZJL/c0\_p0003\_s04.rec

LAB: NULA

REC: DVA

Aligned transcription: D:/HTK/DATA/0004\_MVL/c0\_p0004\_s04.lab vs D:/HTK/DATA/0004\_MVL/c0\_p0004\_s04.rec

LAB: NULA

REC: DVA

====================== 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]

===================================================================

V protokolu se říká, že 44 nahrávek z 50 (88.00 %) bylo rozpoznáno správně.

1. **Testování rozpoznávače naživo**

Dávka **20\_LiveTest.bat**

obsahuje řádek

HVite -H hmm6/hmmdefs -C LiveConfig-FBANK16 -w wordnet -p -70.0 -s 0 dict models0

Je třeba mít připojená sluchátka s mikrofonem k PC.

Dávka se spustí. Uživatel je požádán, aby řekl krátkou větu, na níž si rozpoznávač zkalibruje detektor ticha a řeči. Pak už lze říkat slova a sledovat výsledky.