Pokročilé metody rozpoznávání řeči

Přednáška 8

Rozpoznávání spojité řeči
a jazykové modely

K minulým úlohám

Rozpoznávání mluvčích:

Podmínky: nahrávky od 95 osob, od každé 10 na testování, dalších 90, 45, 20 nahrávek použito na natrénování GMM modelu (jednostavového HMM s více gaussovkami), rozpoznávání v uzavřené sadě

Dosažené výsledky: 100 % úspěšnost dosažena už s GMM s 32 mixturami i pro relativně malý trénovací set (cca 20 nahrávek)

Proč asi?

K minulým úlohám

Detekce klíčových slov:

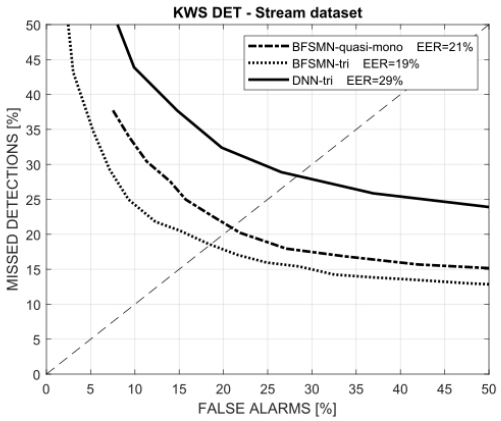
Podmínky: cca 20minutová nahrávka dialogu, relativně spontánní řeč (občas i cross-talk), hledána 2 poměrně dlouhá slova (prezident, komentátor)

Dosažené výsledky: ne příliš dobré - řada výskytů nenalezena, a naopak v řadě případů nalezena jiná slova (a dokonce i hluky)

Proč asi?

Úloha **binární klasifikace/detekce**: dva typy chyb
- false negative (existující slovo nenalezeno – missed detection, MD)
- false positive (nalezeno nesprávné slovo – false alarm, FA)

Obě chyby jdou často proti sobě, jejich poměr závisí na parametru. Kompromis: nastavení parametru do bodu EER (Equal Error Rate)



Základy rozpoznávání spojité řeči (1)

Formulace úlohy:

Máme (zparametrizovanou) nahrávku a chceme určit nejpravděpodobnější sekvenci slov v ní obsažených.

$$P(w_1^*, w_2^*, \ldots w_N^*) = \arg\max_{w} P(W / X)$$

tj. hledáme takovou sekvenci slov ***w****, která je ze všech možných kombinací slov (z daného slovníku ***L***) nejpravděpodobnější, a to vzhledem k sekvenci příznakových vektorů ***X*** = $(x_1, \ldots x_T)$ reprezentujících nahrávku.

Základy rozpoznávání spojité řeči (2)

Modifikace úlohy:

Přímé řešení vztahu z předchozího slajdu není možné, ale můžeme se pokusit úlohu přeformulovat, a to s použitím Bayesova vztahu:

$$P(W/X).P(X) = P(X|W).P(W)$$

a tedy

$$P(W/X) = \frac{P(X|W).P(W)}{P(X)}$$

kde ***P(X|W)*** je pravd. že sekvence slov ***W*** vygeneruje sekvenci ***X***

P(W) je (apriorní) pravděpodobnost sekvence slov ***W***

P(X) je (apriorní) pravděpodobnost sekvence ***X*** (ta je při řešení naší úlohy daná, a tedy konstantní a nemusíme ji uvažovat)

$$P(W/X) \cong P(X|W).P(W)$$

Základy rozpoznávání spojité řeči (3)

Přeformulování a dekompozice úlohy:

Původně jsme hledali:

$$P(w_1^*, w_2^*, \ldots w_N^*) = \arg\max_{w} P(W / X)$$

Nyní budeme hledat

$$P(w_1^*, w_2^*, \ldots w_N^*) = \arg\max_{w,N} P(X \mid W).P(W)$$

kde ***P(X|W)*** je tzv. akustický model (AM, pravděpodobnost že ***W*** vygeneruje ***X)***

P(W) je tzv. jazykový model (LM, pravděpodobnost slovní sekvence ***W)***

První pravděpodobnost už umíme určit pro libovolné slovo, a to pomocí HMM

Druhý člen se naučíme určovat s využitím statistického zpracování textů

Pravděpodobnostní jazykový model (1)

Pro pravděpodobnost sekvence platí (pravidlo o násobení)

$$P(W) = P(w_1, w_2, \ldots w_N)$$
$$= P(w_1)P(w_2 \mid w_1).P(w_3 \mid w_1, w_2)\ldots P(w_N \mid w_1, w_2 \ldots w_{N-1})$$

Většinu členů by bylo obtížné určit, ale můžeme si pomoci aproximací pomocí tzv. N-gramových pravděpodobností (zkráceně N-gramů)

V případě bigramů (N = 2)

$$P(W) \approx P(w_1 \mid start)P(w_2 \mid w_1).P(w_3 \mid w_2)\ldots P(w_N \mid w_{N-1})$$

V případě trigramů (N = 3)

$$P(W) \approx P(w_1 \mid start)P(w_2 \mid w_1, start).P(w_3 \mid w_2, w_1)\ldots P(w_N \mid w_{N-1}, w_{N-2})$$

Pravděpodobnostní jazykový model (2)

N-gramový model: je daný pravděpodobnostmi ***N*** slov za sebou určených ze statistik *trénovacího textového korpusu*

- unigram $p(w_n) = C(w_n) / K$ K ..... pocet vsech slov v trenovacim korpusu

- bigram $p(w_n \mid w_{n-1}) = \frac{C(w_{n-1}, w_n)}{C(w_{n-1})}$

$C(w_{n-1})$ je počet výskytů slova w_{n-1} a
$C(w_{n-1}, w_n)$ je počet výskytů dvojic slov w_{n-1}, w_n.

- trigram $p(w_n \mid w_{n-1}w_{n-2}) = \frac{C(w_{n-2}, w_{n-1}, w_n)}{C(w_{n-2}, w_{n-1})}$

- zerogram $p(w_n) = 1 / L$ všechna slova stejně pravděpodobná (L … velikost slovníku)

Pravděpodobnost sekvence *n* slov vypočítaná z bigramů:

$$P(w_1, w_2, w_3, ...w_n) = p(w_1 \mid start).p(w_2 \mid w_1).p(w_3 \mid w_2)....p(w_n \mid w_{n-1})$$

Pravděpodobnostní jazykový model (3)

Trénování jazykového modelu (bigramového):

1. Nutný je co **největší korpus textů** (všeobecných nebo odborných)
2. Text je třeba předem **vyčistit a normalizovat**.
3. Pro výpočet bigramů je třeba si v paměti **alokovat prostor pro matici četností** a všechny prvky vynulovat.
4. Program prochází text slovo po slovu a za každou nalezenou **dvojici slov ze slovníku** přičte do příslušného prvku 1.
5. Na závěr se určí **bigramové pravděpodobnosti**, a to vydělením součtem četností v řádku (pro stejného předchůdce).
   (Součet pravděpodobností na řádku musí být roven 1).
6. Zbývá vyřešit otázku, co s **nulovými pravděpodnostmi.**
   Pravděpodobnosti bigramů ve větě se násobí, takže jediná nula způsobí, že taková věta nemůže být nikdy správně rozpoznána.
   – řeší se tzv. vyhlazením (smoothing)

Proces odhadu hodnot bigram. LM

1. Slovní páry a jejich četnosti odvozené z korpusu

	from	he	I	often	Paris	to	travel	travels	we	you	Zurich
from					3					2	5
he				2				2		1	
I				2			2				
often	2	1	2		1	4	2	3	1	2	4
Paris	2	1		1		3			1		1
to					4		1			3	2
travel	2	1	1	2		3			1	1	
travels	3	1		2		3				1	
we	1			3			2			1	
you	1		1	3			2				
Zurich	2	1	1	1	1	3			1		
START	1	2	3	1	1		1		3	3	2

Proces odhadu hodnot bigram. LM

1. Slovní páry a jejich četnosti odvozené z korpusu
2. Vypočteny relativní četnosti

	from	he	I	often	Paris	to	travel	travels	we	you	Zurich
from	0	0	0	0	0,3	0	0	0	0	0,2	0,5
he	0	0	0	0,4	0	0	0	0,4	0	0,2	0
I	0	0	0	0,5	0	0	0,5	0	0	0	0
often	0,091	0,05	0,091	0	0,045	0,182	0,091	0,1364	0,045	0,091	0,182
Paris	0,222	0,11	0	0,111	0	0,333	0	0	0,111	0	0,111
to	0	0	0	0	0,4	0	0,1	0	0	0,3	0,2
travel	0,182	0,09	0,091	0,182	0	0,273	0	0	0,091	0,091	0
travels	0,3	0,1	0	0,2	0	0,3	0	0	0	0,1	0
we	0,143	0	0	0,429	0	0	0,286	0	0	0,143	0
you	0,143	0	0,143	0,429	0	0	0,286	0	0	0	0
Zurich	0,2	0,1	0,1	0,1	0,1	0,3	0	0	0,1	0	0
START	0,059	0,12	0,176	0,059	0,059	0	0,059	0	0,176	0,176	0,118

Proces odhadu hodnot bigram. LM

1. Slovní páry a jejich četnosti odvozené z korpusu
2. Vypočteny pravděpodbnosti (jako relativní četnosti)
3. Provedeno vyhlazení – nulové četnosti nahrazeny malými hodnotami

	from	he	I	often	Paris	to	travel	travels	we	you	Zurich
from	0,033	0,03	0,033	0,033	0,233	0,033	0,033	0,0333	0,033	0,167	0,367
he	0,05	0,05	0,05	0,25	0,05	0,05	0,05	0,25	0,05	0,15	0,05
I	0,056	0,06	0,056	0,278	0,056	0,056	0,278	0,0556	0,056	0,056	0,056
often	0,093	0,06	0,093	0,019	0,056	0,167	0,093	0,1296	0,056	0,093	0,167
Paris	0,179	0,11	0,036	0,107	0,036	0,25	0,036	0,0357	0,107	0,036	0,107
to	0,033	0,03	0,033	0,033	0,3	0,033	0,1	0,0333	0,033	0,233	0,167
travel	0,156	0,09	0,094	0,156	0,031	0,219	0,031	0,0313	0,094	0,094	0,031
travels	0,233	0,1	0,033	0,167	0,033	0,233	0,033	0,0333	0,033	0,1	0,033
we	0,125	0,04	0,042	0,292	0,042	0,042	0,208	0,0417	0,042	0,125	0,042
you	0,125	0,04	0,125	0,292	0,042	0,042	0,208	0,0417	0,042	0,042	0,042
Zurich	0,167	0,1	0,1	0,1	0,1	0,233	0,033	0,0333	0,1	0,033	0,033
START	0,068	0,11	0,159	0,068	0,068	0,023	0,068	0,0227	0,159	0,159	0,114

Metody vyhlazování LM (1)

Řeší otázku co s nulovými pravděpodobnostmi (neviděnými dvojicemi)

Metoda ADD1 – ke každému prvku matice se přičte 1

$$p_{+1}(w_n \mid w_{n-1}) = \frac{C(w_{n-1}, w_n) + 1}{C(w_{n-1}) + L}$$

… kde *L* je velikost slovníku (a tedy i počet sloupců v matici)

Metoda jednoduchá, ale nadhodnocuje neviděná slovní spojení

Metody vyhlazování LM (2)

Metoda Witten-Bell – v praxi často používaná

nulové pravděpodobnosti nahrazuje malým číslem, jehož velikost souvisí s tím, kolik má předchůdce **<u>různých</u>** následovníků

$$p_{WB}(w_n \mid w_{n-1}) = \frac{T(w_{n-1})}{Z(w_{n-1})(C(w_{n-1}) + T(w_{n-1}))}.$$

Jestliže $C(w_{n-1}, w_n) > 0$:

$$p_{WB}(w_n \mid w_{n-1}) = \frac{C(w_{n-1}, w_n)}{C(w_{n-1}) + T(w_{n-1})},$$

kde $C(w_{n-1})$ je počet výskytů slova w_{n-1},
$C(w_{n-1}, w_n)$ je počet výskytů dvojic slov w_{n-1}, w_n,
$T(w_{n-1})$ je počet rozdílných dvojic sousedních slov, jejichž první slovo je w_{n-1},
$Z(w_{n-1})$ je počet dvojic sousedních slov, které se neobjevily v trénovacích datech a jejichž první slovo je w_{n-1}.

Příklad:

slovo „mechatronický“ se vyskytuje s několika málo následníky, proto neviděné bigramy dostanou mnohem nižší hodnotu než neviděné bigramy např. u slova „dobrý“

Metody vyhlazování LM (3)

Další často používané metody vyhlazování

Knesser-Ney – metoda podobná dříve uvedené metodě **WB**

Ústupové metody (back-off smoothing)

Princip: pro neviděné n-gramy použijeme n-1-gramy vynásobené vhodným koeficientem

Vyhodnocování kvality LM

Perplexita - nejčastěji používaná míra hodnocení kvality LM

Určuje se na testovacím (neviděném) textu sestávajícím s K slov podle vztahu

$$PP = P(w_1, w_{2,}, \dots w_K)^{-\frac{1}{K}}$$

tedy např. pro bigramový model

$$PP = (P(w_1|start).P(w_2|w_1).P(w_3|w_2) \dots .P(w_{K-1}|w_K))^{-\frac{1}{K}}$$

Čím je hodnota PP nižší, tím je daný LM lepší (má nižší míru neurčitosti).

Pozn.1 Nejvyšší PP bude mít LM, jehož všechny N-gramy budou mít stejnou hodnotu.)

Pozn2. N-gramový model bude mít vždy nižší PP než N-1-gramový.

Nástroje pro vytvoření LM (1)

HTK v základní verzi podporuje práci s bigramy

Nástroje:

HLStats – na daném seznamu textových souborů (vět) a pro daný slovník (seznam slov) spočítá slovní statistiky, zejména unigramy a bigramy a uloží je v souboru bigfn

příklad použití: HLStats -b bigfn -o wordlist labs

HBuild – s využitím slovníku a statistik vytvoří rozpoznávací síť outLatFile

příklad použití: HBuild -n bigfn wordlist outLatFile

Příklad volání rozpoznávače:

HVite -H hmmdefs -S test.scp -i recout.mlf -w outLatFile -p -10.0 -s 0.52 dict models0

Nástroje pro vytvoření LM (2)

Ukázky souborů (slovník bez diakritiky)

Soubor wordlist

```
!ENTER
!EXIT
a
dukle
ho
jsem
kdyz
ker
muzu
rad
schoval
se
tak
ted
uvidel
v
vratit
za
zacinal
ze
```

Soubor labs

```
#!MLF!#
"0.lab"
hezke
odpoledne
vam
v
tehle
chvili
z
regionu
preje
nejen
patrik
rozehnal
ale
take
.
"1.lab"
v
dnesni
```

Soubor bigfn

```
\data\
ngram 1=20002
ngram 2=218038

\1-grams:
-99.999    !ENTER    -0.5444
-1.5219    a         -0.1893
-5.3554    abdikaci  -0.2998
….
\2-grams:
-1.2690    !ENTER a
-4.3255    !ENTER absolutne
-3.2841    !ENTER aby
….
-1.3802    extremne tvrde
-1.3802    extremne vysoka
-1.2041    extremni mira
…
-1.5563    facebook vam
-1.5563    facebook zakazuje
-1.5563    facebook !EXIT
```

Samostatná úloha

Provést experimenty s rozpoznáváním celých vět s použitím existujícího akustického modelu, slovníku a dvou typů jazykového modelu:

- zerogramů
- bigramů natrénovaných na testovacích datech

Návod k řešení (1)

1. Stáhněte si z e-learningu soubor nahrávek (173 nahrávek, v každé 1 věta, od 4 mluvčích)
2. U každé nahrávky najdete kromě souboru WAV také soubor TXT (textový přepis v CP1250) a LAB (textový přepis bez diakritiky).
3. Ze všech slov v textových přepisech si vytvořte **pomocný slovník** a doplňte ke každému slovu výslovnosti (pomocí G2P). U krátkých slov můžete použít i alternativní výslovnosti (např. znělá/neznělá koncová hláska).
4. Z výše uvedeného pomocného slovníku vytvořte HTK slovník **dict** tak, že u všech slovníkových položek **odstraníte diakritiku** (může –> muze ..), převedete **na malá písmena** a (jako obvykle) české symboly fonémů nahradíte anglickými (á –> aa, ..). Na konec slovníku přidáte řádky, které umožní rozpoznávat i neřečové hluky.

```
SILENCE si
SILENCE1 n1
SILENCE2 n2
SILENCE3 n3
SILENCE4 n4
SILENCE5 n5
!ENTER si
!EXIT si
```

Návod k řešení (2)

5. Vytvořte jednoduchou gramatiku typu word-loop, kde každé slovo ve slovníku dict může následovat za jiným (včetně všech hluků typu SILENCE). Tato gramatika odpovídá jazykovému modelu typu **zerogram**. Pomocí HParse vytvořte odpovídající soubor wordnet.
6. Dále si stáhněte soubor **Akusticke_modely**, kde jsou **dva poměrně dobré modely** natrénované na cca 40 hodinách řeči. Jeden má 16 mixtur, druhý 32 mixtur a natrénovány byly 6 iteracemi. První je o trochu horší, ale umožní o trochu rychlejší rozpoznávání.
7. Zparametrizujte si (obvyklým způsobem) testovací nahrávky a proveďte rozpoznávací experimenty, v nichž se budete snažit najít optimální hodnotu –p. Abych vám uspořil práci, vytvořil jsem pro vás soubory testref.mlf a test.scp (upravte si v nich cesty).
8. Lze očekávat hodnoty Acc mezi 70 – 80 % (v závislosti na kvalitě výslovností ve slovníku).

Návod k řešení (3)

9. Pro druhou úlohu jsem vám připravil soubor **s bigramy** outLatFile vytvořený na základě dnešní přednášky. S ním pak můžete provést rozpoznávací experimenty a pokusit se najít optimální hodnoty pro –p a –s (vyrovnávací faktor mezi AM a LM)
10. Lze očekávat hodnoty Acc mezi 90 – 96 % (v závislosti na kvalitě výslovností ve slovníku).

Pozn. Podmínky obou úloh nejsou úplně férové („cheating approach“), neboť
a) slovník je tvořen pouze slovy v testovacích větách
b) bigramový model je naučen přímo na testovacích větách.

Proto lze očekávat nerealisticky vysoké hodnoty úspěšnosti rozpoznávání.

Díky tomu lze ale stanovit jakousi (teoretickou) horní mez, ke které by se mohla přiblížit úspěšnost, kdybychom měli ideální slovník (co největší a s co nejlepšími výslovnostmi) a ideální akustický a jazykový model (natrénovaný na obrovském množství textů)

Výsledky obou úloh prosím opět do konce neděle.