Dialogové systémy

Rozpoznávání řeči

Luděk Bártek

Laboratoř vyhledávání a dialogu, Fakulta Informatiky Masarykovy Univerzity, Brno

jaro 2019

Rozpoznávání řeči

- Rozpoznávání plynulé řeči – převádí souvislou promluvu na psaný text.
- Rozpoznávání izolovaných slov/příkazů.
- Princip rozpoznávání:
  1. získání vektoru příznaků pomocí metod krátkodobé analýzy signálu,
  2. klasifikace na základě vektoru příznaku získaného v předchozím kroku.

Rozpoznávání izolovaných slov

- Slouží k rozpoznání povelů nebo slov (příkazů) zřetelně oddělených na začátku a konci mezerou.
- Odpadá problém stanovení začátku a konce slova v souvislé promluvě.
- Obvykle systémy závislé na uživateli:
  - nutnost natrénování
  - omezená kapacita slovníku.
- Obtíže při rozpoznávání izolovaných slov:
  - Určení začátku a konce promluvy:
    - odlišení šumu od sykavek,
    - detekce nahodilého zvukového vzruchu (klepnutí, ...) kontra okluzívy, které obsahují pauzy,
    - možná přítomnost infrazvuků.
    - ...

Rozpoznávání izolovaných slov
Typy klasifikátorů

- Klasifikátory využívající porovnání slov metodou DTW.
  - Snaží se nalézt co největší shodu mezi rozpoznávaným slovem a slovy v databázi.
- Klasifikátory založené na statistických metodách – modelování pomocí skrytých Markovových modelů:
  - simulace procesu tvorby řeči.
- Klasifikátory pracující na dvou úrovních:
  1. segmentace a fonetické dekódování jednotlivých segmentů
  2. rozpoznání slova na základě dekódovaných segmentů.
- Využití umělých neuronových sítí - více viz:
  - Hinton, O., Teh - A Fast Learning Algorithm for Deep Belief Nets, in Neural Computation, 2006
  - Bengio, L., Popovici, L. - Greedy Layer-Wise Training of Deep Networks, in NIPS' 20016
  - Speech recognition - Lecture 14: Neural Networks

Dynamic Time Warping (DTW)

- Metoda borcení časové osy.
- Používá se pro porovnání dvou číselných řad – dvou úseků promluv (dvou slov).
- Vstup:
  - posloupnost akustických vektorů získaných pomocí metod krátkodobé analýzy signálu
  - databáze akustických vektorů rozpoznávaných slov.
- Výstup – rozpoznané slovo resp. povel.

DTW
Základní postup

- Vytvoříme databázi rozpoznávaných slov (referenční posloupnosti akustických vektorů).
  - Obvykle několik posloupností pro každé slovo, které odpovídají několika způsobům vyslovení příkazu.
- Rozpoznávané slovo převedeme na odpovídající posloupnost akustických vektorů.
- Metodou DTW nalezneme referenční posloupnost akustických vektorů s maximální shodou.

DTW
Formalizace

- Algoritmus DTW hledá parametrizaci f,g:

$$f, g : i = f(k), j = g(k), k \in < 1, K >$$

minimalizující výraz

$$D(A, B) = \sum_{i=1}^{K} d(a_{f(i)}, b_{g(i)})$$

  - d – vzdálenost akustických vektorů (např. Euklidovská metrika)
  - $a_{f(i)}$, $b_{g(i)}$ – referenční a rozpoznávaný příkaz.

DTW
Omezující podmínky

- f,g – neklesající funkce
- Omezení na lokální souvislost a strmost:
  - $0 \leq f(k) - f(k-1) \leq I^*$
  - $0 \leq g(k) - g(k-1) \leq J^*$
  - většinou platí $I^*, J^* = 1, 2, 3$
  - Z praktických testů vyplynulo, že při příliš strmém přírůstku může dojít k nevhodné korespondenci mezi příliš krátkým segmentem vzorku *a* a příliš dlouhým segmentem vzorku *b*.
- Omezení na hraniční body:
  - $f(1) = 1, f(K) = I$, kde I je počet vzorků slova a.
  - $g(1) = 1, g(K) = J$, kde J je počet vzorků slova b.

DTW
Omezující podmínky – pokračování

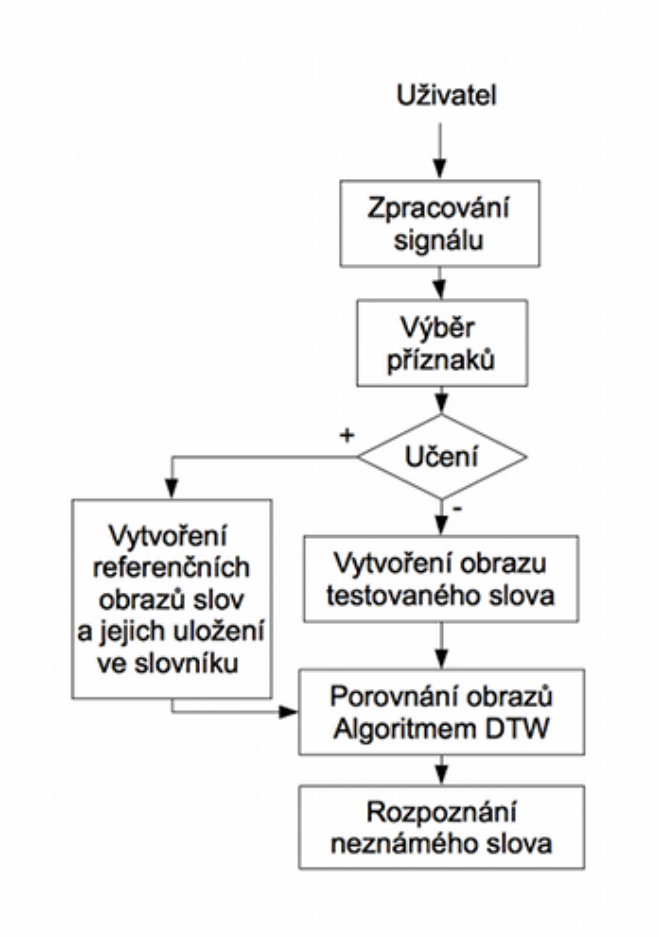
- Globální vymezení oblasti pohybu funkce DTW:
  - omezení minimální a maximální přípustné směrnice přímky vymezující přípustnou oblast pohybu funkce DTW, při splnění podmínky na hraniční body:

$$1 + \alpha[i(k) - 1] \leq 1 + \beta[i(k) - 1]$$

    - α – minimální směrnice přímky omezující přípustnou oblast
    - β – maximální směrnice přímky omezující přípustnou oblast.

DTW – Praktická realizace klasifikátoru slov

Blokové schéma



Obrázek: Blokové schéma klasifikátoru slov

DTW – Praktická realizace klasifikátoru slov
Trénování

- Obecný postup:
  1. Řečník resp. skupina řečníků vysloví postupně každé trénované slovo požadovaného slovníku, buď jednou nebo opakovaně.
  2. Vstupní slova jsou zdigitalizována a následně převedena zvolenou metodou krátkodobé analýzy na posloupnost vektorů příznaků.
  3. Detekce hranic (počátku a konce) slov:
     - Může být náročné na provedení, např. kvůli rušivému pozadí.
     - Nekorektní detekce hranic slov zhoršuje úspěšnost rozpoznávání.
     - Metody odstraňující i jen částečně vliv akustického pozadí zvyšují výpočetní náročnost.
  4. Vytvoření referenčních obrazů slov.

DTW – praktická realizace

Metody vytváření referenčních obrazů slov

- Přímé použití obrazů trénovací množiny jako referenčních obrazů slov – DTW nevyžaduje, aby obrazy téhož slova byly stejně dlouhé, ale z důvodu možnosti aplikace pomocných kritérií, je vhodné provést časovou normalizaci každého obrazu.
- Vytváření průměrného vzorového obrazu pro každou třídu slov *w*:
  - používají se metody lineárního a dynamického průměrování.
- Vytváření vzorových obrazů shlukováním.
  - Vzorové obrazy pro dané slovo se rozdělí do shluků tak, že obrazy uvnitř shluku jsou si ,,podobné“ a obrazy z různých shluků jsou ,,nepodobné“.
  - Shlukování lze realizovat interaktivně (poloautomaticky – metoda řetězové mapy, algoritmus ISODATA), automaticky (algoritmy založené na MacQueenově algoritmu). Více viz závěrečná práce Mgr. Jiřího Kučery.

DTW
Redukce výpočetních a paměťových nároků

- Nevýhody DTW – vysoké paměťové a výpočetní nároky mohou znesnadňovat klasifikaci v reálném čase i při relativně malém slovníku.
- Metody řešení:
  - Hrubá síla – využití paralelních procesorů popř. zákaznických obvodů – může být drahé.
  - Vhodné zakódování parametrů jednotlivých mikrosegmentů referenčních i testovacích obrazů. Využívá se:
    - vektorová kvantizace – počet různých vzorků je konečný – uloží se do kódové knihy a místo hodnoty vzorku se pracuje s jejich indexy v kódové knize.
    - kódová kniha – abeceda všech hodnot, které se vyskytly v signálu (lze kódovat úsporněji než při použití standardního PCM).

DTW
Redukce výpočetních a paměťových nároků

- Využití oblastí spektrální stacionarity – metoda segmentace spektrální stopy.
  - Spektrální stopa – spojnice koncových bodů vektorů příznaků.
  - Lze ji aproximovat – např. lineárními úseky.
- Optimalizace vyhledávání nejbližšího souseda:
  - metody prohledávání metrických prostorů
  - nutno ověřit, že vzdálenost použitá v DTW je metrika.

DTW

Redukce výpočetních a paměťových nároků

- Redukce výpočetních nároků pomocí heuristik při porovnávání.
  - Vícestupňový rozhodovací postup:
    1. porovnání promluvy proti celému slovníku pomocí omezené množiny příznaků
    2. dohledání výsledku kroku 1. pomocí klasického DTW.
  - Práh zamítnutí:
    1. po každém kroku spočítáme vzdálenost slova a obrazu
    2. pokud překročí experimentálně stanovený práh, obraz je zamítnut.

Skryté Markovovské Modely – HMM

- Modelování řeči pomocí HMM vychází z následující představy o tvorbě řeči:
  - Hlasové ústrojí se v krátkém čase nachází v jedné z konečně mnoha artikulačních konfigurací – generuje hlasový signál.
  - Přejde do následující konfigurace.
- Tuto činnost lze modelovat statisticky.
- Kvantizací akustických vektorů lze dosáhnout konečnosti všech parametrů odpovídajícího modelu.

HMM

Principy použití pro rozpoznávání

- Jsou generovány dvě vzájemně svázané časové posloupnosti náhodných proměnných:
  - podpůrný Markovův řetězec – posloupnost konečného počtu stavů
  - řetězec konečného počtu spektrálních vzorů.
- Náhodná funkce ohodnocující pravděpodobnostmi vztah vzorů k jednotlivým stavům.
- Pro rozpoznávání řeči jsou nejčastěji využívané levo-pravé Markovovy modely:
  - vhodné pro modelování procesů spjatých se vzrůstajícím časem.

HMM

Markovův proces

- Markovův proces G se skrytým Markovovým modelem je pětice $G = (Q, V, N, M, \pi)$
  - $Q = q_1, \ldots, q_k$ – množina stavů
  - $V = v_1, \ldots, v_k$ – množina výstupních symbolů
  - $N = (n_{i,j})$ – matice přechodu. Určuje pravděpodobnost přechodu ze stavu q_i v čase t_1 do stavu q_j v čase t_2.
  - $M = (m_{i,j})$ – matice přechodu, určující pravděpodobnost generování akustického vektoru v_j, v kterémkoliv čase ve stavu q_i.
  - $\pi = (\pi_i)$ – vektor pravděpodobností počátečního stavu (pravděpodobnost toho, že stav i je počáteční).
- Trojice $\lambda = (N, M, \pi)$ – vytváří model řečového segmentu.
  - např. Vintsjukův model pro slovo – počet stavů 40 — 50 (odvozeno od průměrného počtu mikrosegmentů ve slově; délka mikrosegmentu 10 ms).

HMM

Určení pravděpodobnosti promluvy

- Značíme $P(O|\lambda)$
- Promluva O standardně zpracována do posloupnosti $O = (o_1, \ldots, o_T)$
  - T – počet mikrosegmentů promluvy
  - o_i – odpovídají výstupním symbolům.
- Určení $P(O|\lambda)$ – metoda využívající rekurzivní výpočet odpředu nebo odzadu generované posloupnosti (forward-backward algorithm).

HMM

Určení pravděpodobnosti promluvy – výpočet

- Výpočet odpředu:
  - α_i – pravděpodobnost přechodu do stavu q_i při generování posloupnosti $\{o_1, \ldots, o_t\}(\alpha_i = P(o_1 \ldots o_t, q_i(t)|\lambda)$
  - Rekurzivní výpočet:
    1. inicializace: $\alpha_1(i) = \pi_i m_i(o_1), i \in < 1, N >$
    2. Rekurzivní krok pro t=1,. . . T-1:

$$\alpha_{i+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) n_{i,j}] m_j(o_{i+1})$$

pro $j \in < 1, N >$, $m(o_t)$ je ekvivalentní zápisu $m_i(l)$, pokud $o_t = v_l$.

    3. Výsledná pravděpodobnost:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

HMM

Alternativní způsob výpočtu $P(O|\lambda)$

- Nevýhoda předchozího postupu:
  - ve výsledném vztahu jsou zahrnuty pravděpodobnosti všech možných posloupností stavů délky T.
- Řešení:
  - výpočet maximálně pravděpodobné posloupnosti stavů Q.
- Výpočet realizován pomocí Viterbiova algoritmu:
  - problém řešen rekurzivně s použitím technik dynamického programování.

HMM

Trénování parametrů modelu $\lambda = (N, M, \pi)$

- Nutno stanovit postup při trénování parametrů modelu.
- Cíl trénování:
  - maximalizace pravděpodobnosti $P(O|\lambda)$
- Problém:
  - neexistuje analytická metoda ke zjištění globálního maxima funkce n proměnných.
- Řešení:
  - lze použít iterativní algoritmy zajišťující aspoň lokální maximalitu.
- Nejpoužívanější postup – Baum-Welchův algoritmus.
- Další problém při trénování modelu:
  - vliv konečné trénovací množiny:
    - čím menší trénovací množina a čím větší matice M, tím větší pravděpodobnost, že některé prvky zůstanou nastaveny na 0 (problém chybějících/neadekvátních dat).

HMM

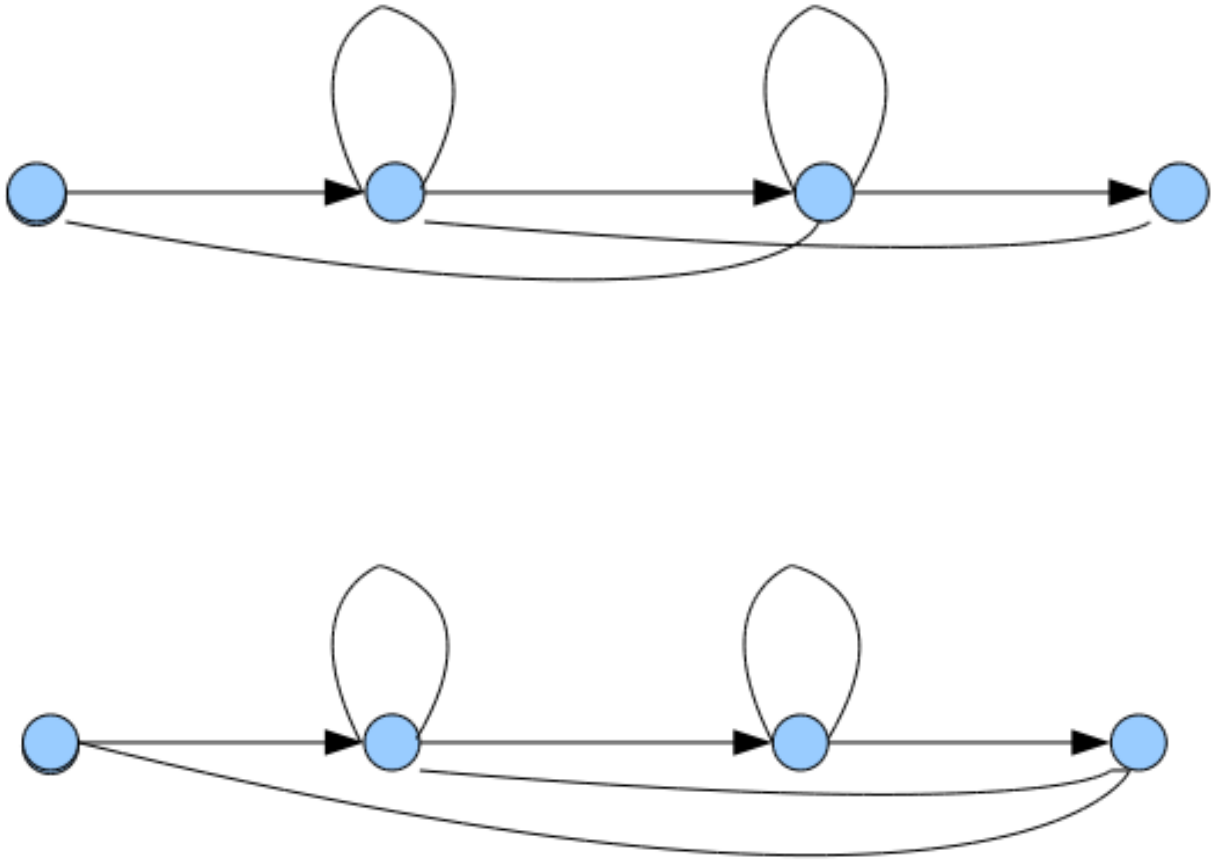
Rozhodovací pravidlo při rozpoznávání izolovaného slova

- Používá se princip maximální věrohodnosti.
  1. Pro slovo O a všechna λ:
     1. Spočítáme $P(O|\lambda)$.
  2. Jako výsledek vybereme třídu s maximální hodnotou $P(O|\lambda)$.

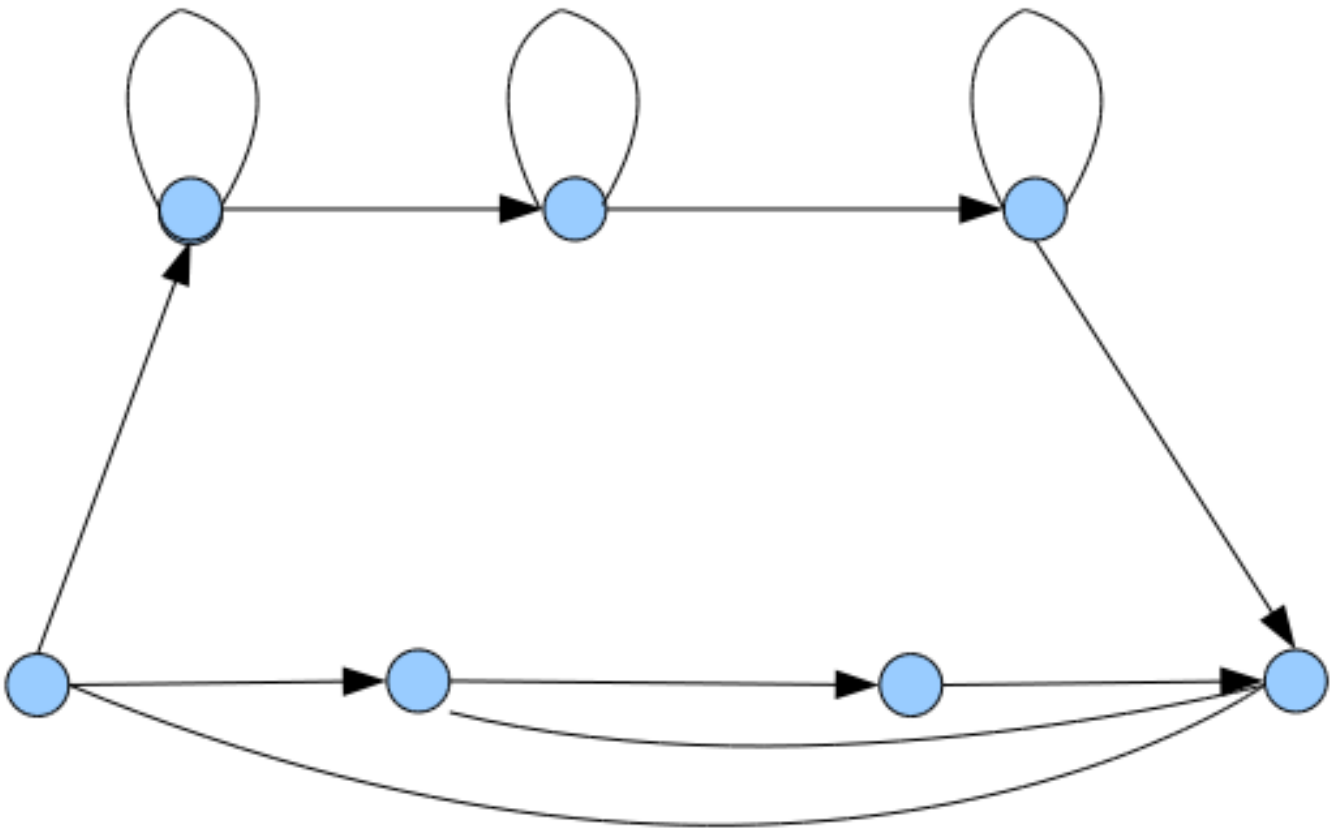
HMM
Implementace

- Modelování povelů:
  - nejčastěji se používají modely se 4 — 7 stavy.
  - Pro modelovaní lze využít nástroje pro tvorbu HMM
    - HTK – Hidden Markov Model Toolkit.
- Modelování fonémů:
  - obvykle 4 — 7 stavů
  - model slova – zřetězení modelů fonémů
  - problémy s výpočtem v reálném čase
    - lze řešit pomocí speciálních algoritmů pro hledání maxima $P(O|\lambda)$.

Příklady struktur pro fonémy



Příklady struktur pro fonémy



Rozpoznávání plynulé řeči

- Hlavní rozdíly oproti rozpoznávání slov:
  - nelze vytvořit databázi vzorů
  - nutno brát zřetel na prozodické faktory
  - nutno určovat hranice mezi slovy
  - vypořádání se s výplňkovými zvuky a chybami řeči.
- Řešení – statistický přístup:
  - jazykový model
  - model uživatele.
- Příklad: HMM vrátí stejnou pravděpodobnost např. pro slova ,,máma“ a ,,nána“ – nejspíše se použije máma – je častější.

Rozpoznávání plynulé řeči

Jazykové modely

- Máme:
  - posloupnost slov (promluva) $W = (w_1, \ldots, w_n)$
  - posloupnost akustických vektorů $O = (o_1, \ldots, o_t)$.
- Chceme nalézt W^* (množinu všech promluv), která maximalizuje $P(W|O)$.
- Dle Bayesova pravidla platí:

$$P(W^*|O) = max\, P(W|O) = max\, \frac{P(W) * P(O|W)}{P(O)}$$

Rozpoznávání plynulé řeči

Jazykové modely – pokračování

- Pro nalezení maxima potřebujeme znát:
  - model řečníka – $P(O|W)$
  - jazykový model – $P(W)$.
- Model řečníka lze nahradit pravděpodobností generování W odpovídajícím Markovovým modelem.
- Trigramový model:
  - Experimentálně ověřeno, že platí:

$$P(w_n|w_1 \ldots w_{n-1}) \cong P(w_n|w_{n-2}w_{n-1})$$

.

Rozpoznávání plynulé řeči
Rozpoznávání tématu

- Úspěšnost rozpoznávání řeči se pohybuje cca 50 % — 99 % v závislosti na úkolu, jazyku, . . .
- Úspěšnost rozpoznávání lze zvýšit omezením domény rozpoznávání:
  - rozpoznání tématu
  - použitím gramatik pro rozpoznávání řeči.
- Známé téma:
  - změna stavového prostoru a pravděpodobnosti trigramů:
    - např. burzovní zprávy – rozpoznáno ,,honey“ nebo ,,money“?
  - možnost vytvoření přesnějšího jazykového modelu.

Gramatiky pro podporu rozpoznávání řeči

- Úspěšnost obecného rozpoznávání plynulé řeči může klesnout až na cca 50 %.
- Zvýšení lze dosáhnout omezením domény – např. specifikováním přípustných vstupů.
- Lze použít gramatiky pro podporu rozpoznávání řeči:
  - bezkontextové gramatiky
- Způsoby zápisů gramatik:
  - prostředky logického programování
  - proprietární řešení
  - otevřené standardy – JSGF, W3C SRGS, ...

Gramatiky pro podporu rozpoznávání řeči
Java Speech Grammar Specification (JSGF)

- Textový zápis gramatiky nezávislý na platformě a prodejci.
- Určen pro použití při rozpoznávání řeči.
- Součást Java Speech API.
- Používá styl a konvence jazyka Java.
- Aktuální verze 1.0 (říjen 1998).
- Použit např. v rozpoznávači Sphinx-4, VoiceXML interpretru VoiceGlue, ...
- Podrobněji v 2. polovině semestru při probírání tvorby dialogových rozhraní.

Gramatiky pro podporu rozpoznávání řeči
Ukázka JSGF


```
#JSGF
<koren> = Chci jet <cim> .|
Chci jet <cim> z <odkud> do <kam> .|
Chci jet <cim> z <odkud> do <kam> v <kdy> .;
<cim> = vlakem | autobusem;
<odkud> = <czMesto>;
<kam> = <czMesto>;
<kdy> = <czCas>;
```

Gramatiky pro podporu rozpoznávání řeči
W3C Speech Recognition Grammar Specification (SRGS)

- Standard W3C.
- Aktuální verze 1.0 (březen 2004).
- Definuje způsob zápisu pravidel a jejich odkazování.
- Dva způsoby zápisu:
  - XML
  - ABNF (Augmented BNF).
- Podrobněji v 2. polovině semestru při probírání tvorby dialogových rozhraní.

Ukázka W3C SRGS


```
#ABNF 1.0 UTF-8
root $pozdrav;
language cs-CZ;
mode voice;
$pozdrav = ahoj

<?xml version="1.0" encoding="utf-8" ? >
<grammar root="pozdrav" xml:lang="cs-CZ" version="1.0" >
<rule id="pozdrav" >
ahoj
< /rule>
< /grammar>
```