Kapitola 26

Státnice I3: Analýza a syntéza mluvené řeči

26.1 Speech Recognition

- The task of transforming of acoustic signal into text.
- Based on statistical methods
- Constant increase of performance over last decade
- Parameters influencing the recognizers performance:
  - size of the vocabulary
    - yes/No recognition
    - digit recognition — 10 words
    - large vocabulary — 20,000 to 60,000 words
  - fluency of speech
    - isolated words recognition
    - continuous speech recognition
  - noise to signal ratio
- The following text will focus on **Large-Vocabulary Continuous Speech Recognition** (although the methods are applicable universally), the methods shown are **speaker independent** (The system was not trained on the person to be recognized)

Speech Recognition Architecture

- **Noisy-channel paradigm**
- Acoustic input $O = o_1, o_2, o_3, ..., o_t$
  - consist of individual "acoustic observations" o_i
    - o_i is represented as a feature vector
    - usually one acoustic observation for each 10ms
- Sentence treated as a string of words $W = w_1, w_2, w_3, ..., w_n$
- Probability model:

$$W^* = \arg\max_{W \in L} P(W|O) = \arg\max_{W \in L} \frac{P(O|W)P(W)}{P(O)} = \arg\max_{W \in L} P(O|W)P(W)$$

- $P(W)$ — the prior probability — computed by **language model**
- $P(O|W)$ — the observation likelihood — computed by **acoustic model**

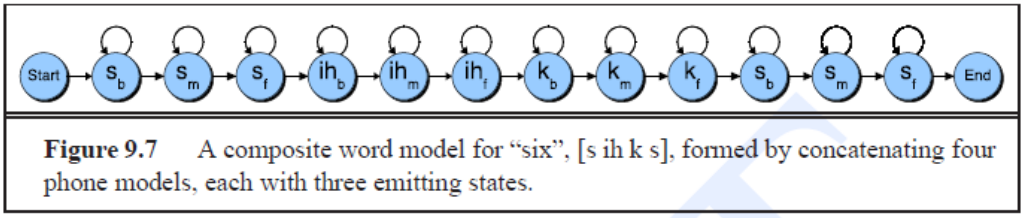


Figure 9.7 A composite word model for "six", [s ih k s], formed by concatenating four phone models, each with three emitting states.

Obrázek 26.1: (Speech and Language Processing(draft) — Jurafsky, Martin)

The Hidden Markov Model Applied To Speech

Vezmeme Markovův model, kde skryté stavy odpovídají hláskám (nebo jejich částem) a emitované znaky odpovídají akustickému pozorování podle akustického modelu. HMM pro analýzu řeči má ale speciální strukturu:

- Typical feature of ASR HMMs: **left-to-right** HMM structure
  - HMM don't allow transition from states to go to earlier states in the word
  - states can transition to themselves or to successive states
    - transitions from states to themselves are extremely important — durations of phones can vary significantly (for example: duration of [aa] phone varies from 7 to 387 miliseconds — 1 to 40 frames)

HMM representation of phone

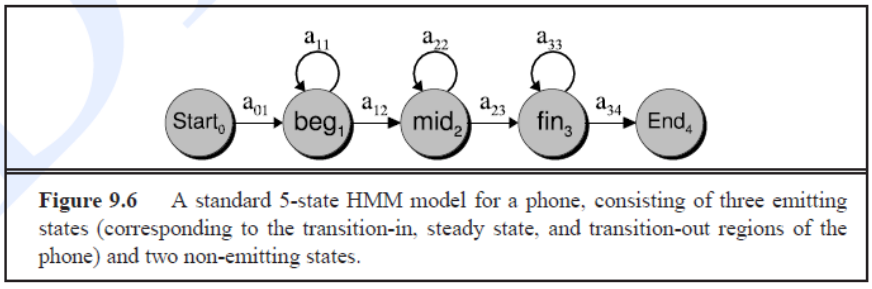


Figure 9.6 A standard 5-state HMM model for a phone, consisting of three emitting states (corresponding to the transition-in, steady state, and transition-out regions of the phone) and two non-emitting states.

Obrázek 26.2: (Speech and Language Processing(draft) — Jurafsky, Martin)

- Figure 9.6
  - Phones are non-homogeneous over time
    - Thus there are separate states modelling a beginning, middle, and end of each phone.

HMM representation of word

- pronounciation lexicon needed — tell us for each word what phones it consists of (sometimes multiple variants of pronounciation)
  - pronounciation lexicon of English: CMU dictionary (publicly available)
- Concatenation of HMM representations of phones

Speech recognition HMM

- Figure 9.22:
  - Combination of word HMMs
  - adde special state modelling silence
  - transition from end state to start state — sentence can be constructed out of arbitrary number of words
  - transitions from the start state are assigned unigram LM probabilities, higher n-grams also possible

- Figure 9.29:

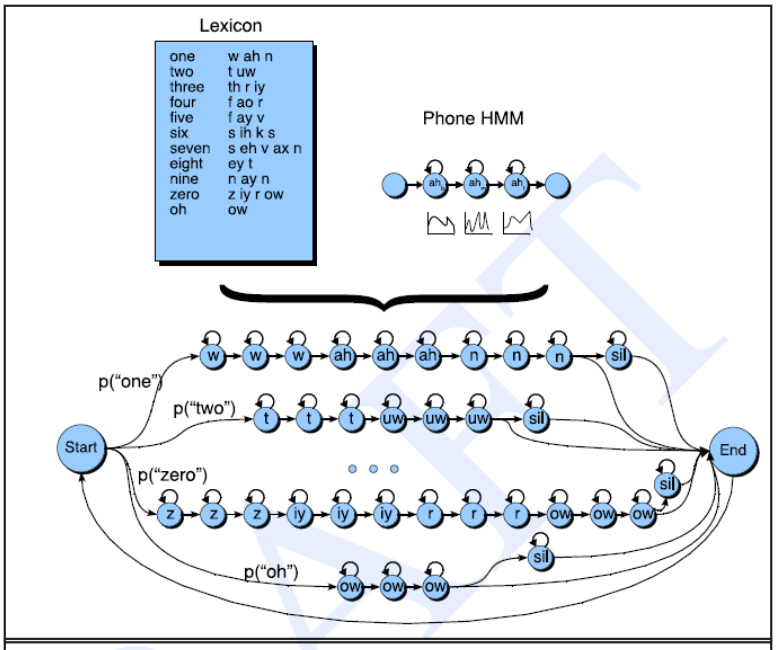


Figure 9.22 An HMM for the digit recognition task. A lexicon specifies the phone sequence, and each phone HMM is composed of three subphones each with a Gaussian emission likelihood model. Combining these and adding an optional silence at the end of each word, results in a single HMM for the whole task. Note the transition from the End state to the Start state to allow digit sequences of arbitrary length.

Obrázek 26.3: (Speech and Language Processing(draft) — Jurafsky, Martin)

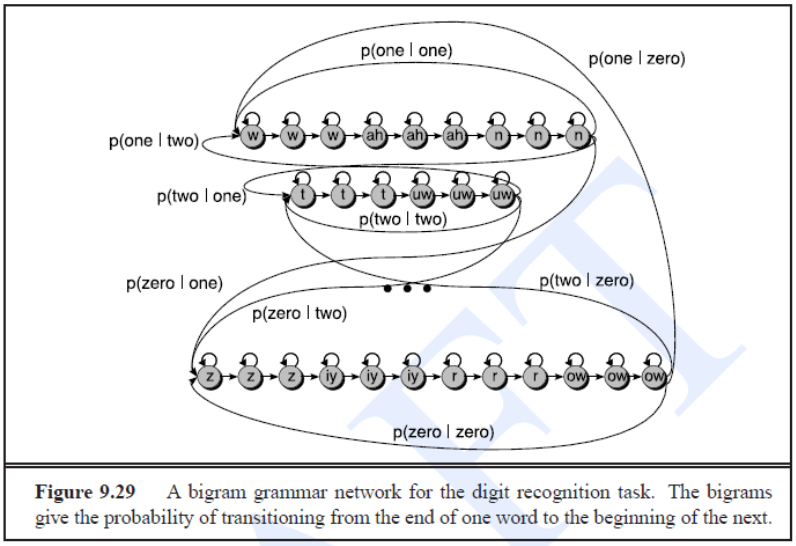


Figure 9.29 A bigram grammar network for the digit recognition task. The bigrams give the probability of transitioning from the end of one word to the beginning of the next.

Obrázek 26.4: (Speech and Language Processing(draft) — Jurafsky, Martin)

  - start state, end state and silence states omitted here for a convinience reason
  - Bigram language model used here — probabilities of transitions from the ends to the beginnings of the words

Feature Extraction: MFCC Vectors

- MFCC — mel frequency cepstral coefficients — most common feature representation (– spectral features)

Analog-to-Digital Conversion

- **sampling**
  - measuring of the the amplitude of the signal at a particular time
  - sampling rate — number of samples per second
  - maximum frequency that can be measured is half of the sampling rate
- **quantization**
  - Represenation of real-valued numbers (amplitude measurements) as integer
    - 8 bits => values from -128 to 127
    - 16 bits => values from -32768 to 32767

Preemphasis

- boosting of amount of energy in the high frequencies
- improves phone detection accuracy – vyšší harmonické frekvence jsou pak znatelnější

Windowing

- spectral features are to be extracted from a small segments of speech
  - asumption that the signal is stationary on small segment
- (rougly) stationary segments of speech extracted by using a **windowing** technique
- windowing characterized by
  - window's **width**
  - **offset** between succesiev windows
  - **shape** of the window
- segments of speech extracted by windowing are called **frames**
  - **frame size** — number of miliseconds in each frame
  - **frame shift** — miliseconds between the left edges of successive windows

Discrete Fourier Transform (DFT)

- extracts spectral information for the windowed signal
- how much energy each frame contains at different frequency bands

Mel Filter Bank and Log

- human ears less sensitive to higher frequencies (above 1000Hz)
- human hearing is "logaritmic"
  - (i.e. for human, distance between 440Hz and 880Hz equals the distance between 880Hz and 1760Hz — in both examples, the distance is one musical octave)
- these feature of hearing are utilized in speech recognition
- DFT outputs are warped onto the **mel scale** $\text{mel}(freq) = 1127 \ln(1 + \frac{freq}{700})$
  - Frequency scale divided into bands (frekvenční pásma)
    - 10 bands linearly spaced below 1000Hz

    - remaining bans spread logarithmically above 1000Hz
  - new **mel-scaled** vectors
    - energy collected from each frequency band
    - logarithms of energy values — the human response to signal level is logarithmic

The Cepstrum

briefly:

- inverse DFT of the MEL scaled signal
- result — (usually 12) cepstral coefficients for each frame
- motivation — these coefficients are uncorelated, "they model the vocal tract better"

Deltas and Energy

- So far — 12 cepstral features
- 13th feature — **Energy**: obtained by suming squares of signal energy for all samples in the given frame
- For each feature:
  - **delta** cepstral coefficient obtained as a derivation of feature values, computed as

$$d(t) = \frac{c(t+1) - c(t-1)}{2}$$

    (for particular cepstral value c(t) at time t)
  - **double delta** cepstral coefficient obtained as a derivation of **delta** cepstral feature values

MFCC Summary

- 39 MFCC features

```
12 cepstral coefficients
12 delta cepstral coefficients
12 double delta cepstral coefficients
1 energy coefficient
1 delta energy coefficient
1 double delta energy coefficient
```

Acoustic Likelihood Computation

- Last chapter — how to obtain feature vectors
- This chapter — how to compute likelihood of these feature vectors given an HMM state
  - i.e. how to obtain the **emission probabilities**

$$B = b_i(o_t) = p(o_t|q_i)$$

Vector Quantization

- simple method, not used in state of the art systems
- idea: map feature vectors into a small number of classes
- **codebook** — list of possible classes
- **prototype vector** — feature vector representing the class
- codebook is created by **clustering** of all the feature vectors in the training set into the given number of classes
- prototype vector can be chosen as a central point of each cluster
- each incoming feature vector is compared to all prototype vectors, the closest prototype vector is selected, and the feature vector is replaced by the class label of the selected prototype vector.
- disadvantage of this method: loss of specific information about the given feature vector, significant impact on performance
- advantage: emissions probabilities can be stored for each pair of HMM state and output symbol, Baum-Welch training conceptually easier

Gaussian Probability Density Functions

- more adequate method for modelling of emission probabilities
- used in state of the art systems
- for each HMM state, emission probability distribution over the space of possible feature vectors is expressed by the **Gausian Mixture Model** (GMM) (tj. jde o složení gaussovských funkcí, každá odpovídá skrytému stavu a je charakterizovaná střední hodnotou a rozptylem, ty se získávají z korpusu):

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm} \frac{1}{\sqrt{2\pi|\Sigma_{jm}|}} exp[(o_t - \mu_{jm})^T \Sigma_{jm}^{-1}(o_t - \mu_{jm})]$$

μ_{jm} - mean of the "m-th" Gaussian of the state "j"

$$\Sigma jm$$

— covariance matrix of the m-th Gaussian of the state j

- Estimation of the GMM parameters (mean and covariance matrix) — **Baum-Welch algorithm**

Search and Decoding

- Bayes probability formula:

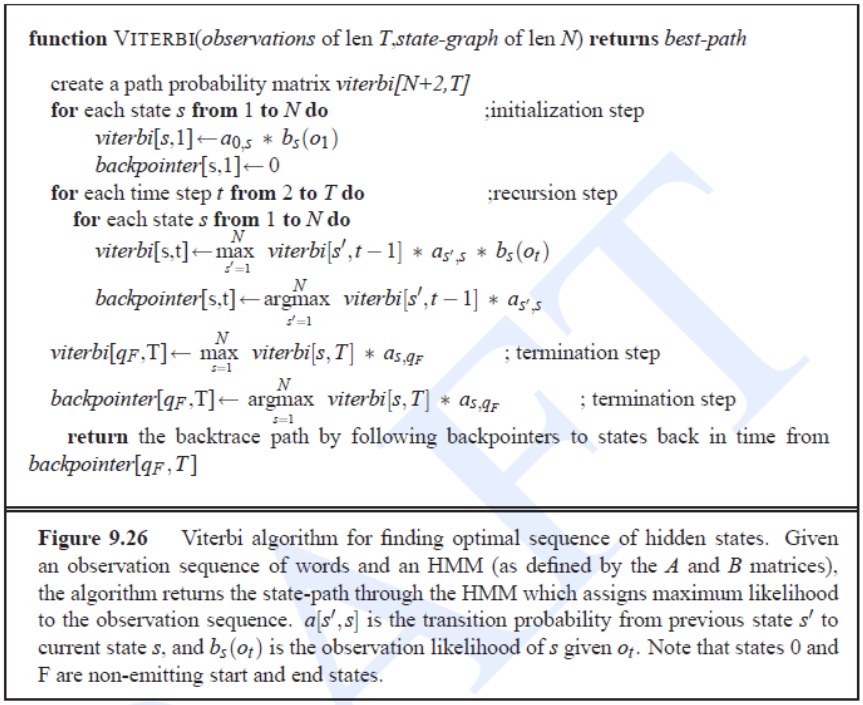
$$W^* = \arg\max_{W \in L} P(O|W)P(W)$$

- typically, more complex formula used:

$$W_* = \arg\max_{W \in L} P(O|W)P(W)^{LMSF} WIP^N$$

  - LMSF — language model scaling factor (kvůli tomu, že jednotlivé framy v $P(O|W)$ nejsou nezávislé, musíme provést vyvážení obou bayesovských komponent)
  - WIP word insertion penalty (vážení jazykového modelu má vedlejší efekt – snižuje penalizaci za příliš mnoho slov ve větě, to je nutné kompenzovat)
  - N — number of words in sentence

- decoding — **Viterbi algorithm**
  - finds the most probably sequence of HMM states
  - the output sentence can be easily constructed out of this sequence of HMM states
  - **beam search pruning**
    - at each trellis stage, compute the probability of best state/path D. prune away any state that is less probable than $D \times \Theta$, where Θ is the beam width (value lower than 1)



function VITERBI(*observations* of len *T*,*state-graph* of len *N*) **returns** *best-path*

create a path probability matrix *viterbi*[*N*+2,*T*]
for each state *s* **from** 1 **to** *N* **do** ;initialization step
  viterbi[*s*,1] $\leftarrow a_{0,s} * b_s(o_1)$
  backpointer[s,1] $\leftarrow 0$
for each time step *t* **from** 2 **to** *T* **do** ;recursion step
  for each state *s* **from** 1 **to** *N* **do**
    viterbi[s,t] $\leftarrow \max_{s'=1}^{N}$ *viterbi*$[s',t-1] * a_{s',s} * b_s(o_t)$
    backpointer[s,t] $\leftarrow \operatorname{argmax}_{s'=1}^{N}$ *viterbi*$[s',t-1] * a_{s',s}$
viterbi[q_F,T] $\leftarrow \max_{s=1}^{N}$ *viterbi*$[s,T] * a_{s,q_F}$; termination step
backpointer[q_F,T] $\leftarrow \operatorname{argmax}_{s=1}^{N}$ *viterbi*$[s,T] * a_{s,q_F}$; termination step
  return the backtrace path by following backpointers to states back in time from *backpointer*[q_F,T]

Figure 9.26 Viterbi algorithm for finding optimal sequence of hidden states. Given an observation sequence of words and an HMM (as defined by the A and B matrices), the algorithm returns the state-path through the HMM which assigns maximum likelihood to the observation sequence. $a[s',s]$ is the transition probability from previous state s' to current state s, and $b_s(o_t)$ is the observation likelihood of s given o_t. Note that states 0 and F are non-emitting start and end states.

Obrázek 26.5: (Speech and Language Processing (draft) — Jurafsky & Martin)

Embedded Training

- Recall: A — HMM transition probabilities, B — emission probabilities (modeled by GMMs)
- Given: phoneset, pronunciation lexicon and the transcribed wavefiles:
  - Build a whole sentence HMM for each training sentence (concatenation of HMM for distinct words)
  - Initialize A probabilities to 0.5 (for loop-back or for the correct next subphone) or to zero (for all other transitions)
  - Initialize B probabilities by setting the mean and variance for each Gaussian to the global mean and variance for the entire training set
  - Run multiple iterations of the Baum-Welch algorithm
- This process will optimize the A and B probabilities
  - i.e. for each HMM state corresponding to a distinct subphone (each phone modelled by 3 subphones), the probability of loop-back transition and leaving transition is reestimated as well as the parameters of GMM for the given state.
  - phone is modelled by the same parameters independently from which word it occurs in (i.e. same HMM for phone [a] in every word containing [a] phone) — "(this is not written anywhere in Jurafsky & Martin, but it seems to be intuitive and it's hopefully true:))

Evaluation: Word Error Rate

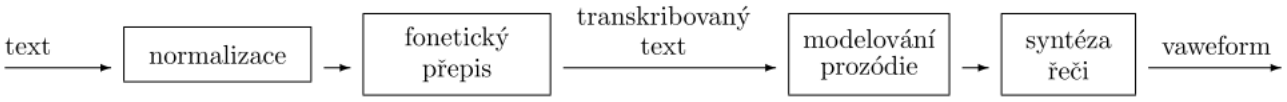
- **word error rate**: based on how much the word string returned by the recognizer differs from a reference transcription
- based on **minimum edit distance** in words between the hypothesized and correct string (number of **substitutions,** insertions **and** deletions"' needed to map between these sentences)

$$\text{Word Error Rate} = 100 \times \frac{\text{Insertions} + \text{Substitutions} + \text{Deletions}}{\text{Total Words in Correct Transcript}}$$

26.2 Speech Synthesis

Úvod

Základní schéma Text-To-Speech systému



- Text je většinou nějaký spec. případ, např. e-maily atp., text i transkripce většinou v nějaké formě obsahuje značky pro suprasegmentální fonémy.
- Syntéza řeči je jen jedna z částí TTS systému, to je nutné rozlišovat. Jde možná o nejsložitější a nejdůležitější součást, ale její vstup není text.
- Normalizace je někdy nutá provádět, někdy ne. Obsahuje např. vyházení hlaviček z e-mailů, příp. přidání spec. prozódie pro ně atd. Používá se, aby stejný TTS systém mohl být použit k různým věcem (předpřipravení textu podle jeho očekávaného typu).
- Grafém, písmeno, letter je nejmenší jednotka psané podoby jazyka. S některými jsou problémy, zda je považovat za jediný grafém (např. písmena s diakritikou), ale nám to většinou bude jedno.
- Hláska, sound je nejmenší jednotka zvukové podoby jazyka.
- Foném je nejmenší strukturální jednotka zvukové podoby jazyka, která rozlišuje význam.
- Fonetický přepis je přepis zvukové podoby textu, zaznamenávající hlásky, příp. suprasegmentální jevy. Může být postavená na pravidlech nebo na slovníku (ale vždy se používají oba komponenty, jeden slouží jako doplněk). Největší problém dělá přepis v jazycích, které např. neznačí samohlásky.

Fonetika a fonologie

Potřebujeme jednak popis výslovnosti, jednak popis akustiky (zvukových vln, jimiž se jednotlivé hlásky projevují). Můžu mít i popis vnímání (percepce) hlásek, ale ten pro naše účely není nezbytný.

Hlavní rozdíl mezi fonetikou a fonologií je ten, že fonetice jde o víceméně fyzikální akustický popis všech zvuků a hlásek, kdežto fonologii zajímá systém, struktura jazyka. Pro fonologický výzkum narozdíl od fonetiky potřebujeme alespoň nějaké základní informace o daném jazyce.

Akustika

Jednotlivé hlásky jsou složené zvuky (vlnění vzduchu), obsahující tónové (periodické) a šumové (neperiodické) složky. Rozlišujeme je právě podle složení jejich zvuku, např. konsonanty mají větší podíl šumových a vokály tónových složek, konsonanty se dále liší svou znělostí či neznělostí jako přítomností tónových složek.

Hlasové ústrojí se dá zjednodušeně představit jako zdroj zvuku (hlasivky) a rezonanční prostor (nadhrtanové dutiny). Hlasivky kmitají na nějaké základní frekvenci (F_0) a v nadhrtanových dutinách se zesilují některé harmonické frekvence.

Zvuk můžeme rozložit na frekvenční spektrum a zkoumat sílu zastoupení jednotlivých frekvencí v čase. Provádí se to běžně na počítači pomocí Fourierovy analýzy, výsledkem je spektrogram.

Výrazně zesílené frekvence číslujeme F_1, F_2 atd., a nazýváme formanty. Prvních několik je zastoupených v signálu "úmyslně", vlivem výslovnosti, příliš vysoké frekvence ale už člověk neovlivní. Proto při záznamu zvuku snímáme jen úzké pásmo (např. do 5 nebo 9 kHz). Frekvence, které jsou na spektru zvuku "úmyslně" potlačeny, nazýváme antiformanty.

U konsonantů (hlavně u neznělých) většinou nehledáme formanty, ale transienty – jde o "přechody" ve spektru, na místě, kde začínají nebo končí formanty okolních samohlásek. Místo, kam transienty ukazují (tedy hypotetický bod na spektru) pro danou souhlásku nazýváme locus. Bod locu je důležitý pro rozpoznání neznělých hlásek, považuje se za centrum jejich šumu.

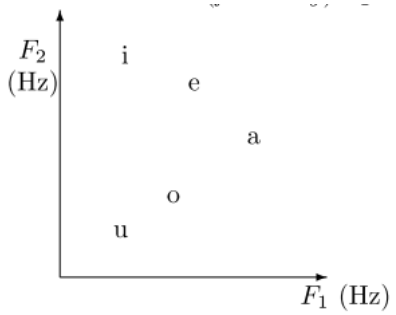
Samohlásky mají nejvýraznější formanty ($F_1, F_2, \ldots$). Pokud není u nějaké hlásky přítomna F_0 (a tedy ani výrazné vyšší formanty), jedná se o neznělou hlásku. Podle šumu (nekoncentrovaný signál po velké části spektra) se poznají frikativy (šumové hlásky). Explozivy (závěrové hlásky) se poznají okamžikem ticha a následným šumem exploze.

Fonetická abeceda IPA

Abeceda IPA slouží pro fonetickou transkripci, hlavně v jazykově nezávislém prostředí. Některé jazyky ji používají i pro zápis své výslovnosti, v některých se používají jiné abecedy, protože jsou pro ně šikovnější (např. v češtině). Hlásky, kterým je přiřazena jedna značka, se mohou napříč jazyky lišit – jde jen o aproximaci. Pro odlišení hlásek v rámci jednoho jazyka ale většinou plně dostačuje.

Vokalický systém

Extremální vokály jsou [i] (jazyk je nahoře vepředu), [u] (nahoře vzadu), [a] (dole uprostřed). Některé zvuky si v různých jazycích více či méně odpovídají, některé jazyky rozlišují více vokálů než jiné. Rozlišení může být podle několika různých vlastností najednou (a některé jejich kombinace nemusí být povolené).



Obrázek 26.6: První dva formanty českých samohlásek. Všimněte si, že zrcadlí místo jejich tvoření.

Pro akustický popis používáme tak krátké zvuky, že se nezmění pozice jazyka ("statický zvuk"), ale už se můžou analyzovat frekvence, ze kterých se zvuk skládá. Zajímáme se o lokální maxima frekvencí. Tím nalezneme nejnižší lokální maximum – základní frekvenci F_0 – a některé její vyšší harmonické frekvence. Podle prvních dvou formantů F_1 a F_2 lze samohlásky dobře odlišit (viz obrázek).

Konsonantický systém

Základní dělení konsonantů je na znělé, voiced a neznělé, voiceless. Ty se liší přítomností základního hlasového tónu (tj. kmitáním hlasivek při jejich tvorbě).

Podle místa artikulace rozlišujeme mj. labiály (obouretné) hlásky [p, b, m], labiodentály (retozubné) [f, v], dentály (zubné) [θ, ð], prealveolární [t, d, s, z, ts, dz], postalveoláry [š, ž], palatály (tvrdopatrové) [ď, ť, ň], veláry (měkkopatrové) [k, g, ch], glotály (hlasivkové) [h].

Podle způsobu tvoření rozlišujeme plozivy, závěrové [p, b, t, d, k, g, ť, ď], nasální, nosové plozivy [n, m, ŋ], frikativy, šumové [s, z, š, ž, ch, f, v], afrikáty, polozávěrové [ts, tš, dz, dž], vibranty, trills [r], bokové hlásky, laterály [l], aproximanty [j, w] (ty se vlastně fyzicky neliší od samohlásek, jde jen o popis).

Pro akustický popis konsonantů jsou určující transienty a bod locu. Locus se dá zhruba odhadnout podle místa tvoření – čím zadnější hláska (čím blíž je místo tvoření hlasivkám), tím vyšší je locus.

Pro nosovky je charakteristický nasální komponent na frekvenci cca 200-300 Hz (tedy pro vysoké hlasy nevýrazný) a potlačení formantu F_1 (vzniká antiformant).

Prozódie

Prozódie zahrnuje všechny vlastnosti, které se projevují nad hranicemi segmentů. Sestává z:

- F_0 základní tón hlasu, voice pitch
- časování, timing
- intenzity (není totéž, co hlasitost – je měřitelná, hlasitost je dojem; pro TTS ale není tak důležitá)

Vždy tu pracujeme jen s relativními hodnotami a prominencí (zvýrazněním) v některé z nich.

Hlásky ve skutečnosti existují až ve slabikách. Slabika je nejmenší část mluveného textu, která se dá zopakovat izolovaně – konsonant je vždy závislý na vokálu své slabiky a naopak. Vnímání slabik je jazykově závislé: když definuju slabiku jako "peak in sonority", bude slovo "lžu" sestávat ze 2 slabik.

Vyšší jednotka je přízvukový takt, fonologické slovo, stress unit. To je skupina slabik, z nichž na jedné je přízvuk. Přízvuk závisí na konkrétním jazyku a jedná se o kombinaci timingu, intonace i intenzity (prominence v některé z těchto hodnot).

Nad úrovní slov rozlišujeme intonační jednotky, intonation contours. Ty jsou relativně nezávislé, mezi nimi má člověk tendenci dělat pauzu v řeči. Jejich rozlišení ale není úplně přesné.

Nejvyšší jednotkou je celé vyjádření, utterance. Např. v dialogu odpovídá větě, ale může být i delší. Finální intonace vyjádření je terminální.

Melodie má v některých jazycích distinktivní funkci – stejné slabiky s jinou melodií mají jiný význam. Takové jazyky se nazývají tónové. Větná melodie ale může mít zároveň jinou funkci.

Mikroprozódie zahrnuje všechno, co se děje v rámci jedné hlásky, ale je ovlivňováno okolím. Má vliv na velkou prozódii. Je podvědomá, záleží i na konkrétních hláskách. Mikroprozodickým fenoménem je např. délka hlásky (záleží ale na tom, jestli délka hlásky rozlišuje význam). Další je např. změna tónu hlasu v rámci jedné hlásky. Běžný TTS systém se mikroprozodií nezabývá, protože ji má nahranou ve svém korpusu segmentů; prozódií se ale zabývat musí.

Stavba Text-To-Speech systému

Normalizace

Někdy se dohromady s normalizací textu dělá chunking, tj. rozdělení textu na dostatečně malé kousky pro zpracování, někdy je jako samostatný krok. Data se musí rozdělit někde, kde je to možné (ne např. uprostřed věty).

Fonetický přepis

Jde o přepis letter-to-sound, tedy přepisujeme grafémy na hlásky. Odlišují se dva přístupy:

- založený na pravidlech, rule-based
- založený na slovníku, dictionary-based

V dnešních systémech jsou v podstatě vždy přítomna i pravidla i slovník, ale jedna metoda je vždy primární, druhá doplňková.

Pro pravidla se de facto dají použít regulární výrazy, tj. přepisy se zapojením kontextu na obě strany. Většinou neoperují přímo nad textem, ale nad nějakými speciálně vytvořenými datovými strukturami.

Některými pravidly musí dojít k zjednoznačnění (disambiguization) textu, např. různou interpunkci apod. je nutné správně interpretovat – tečka např. může mít spoustu významů. Někdy se víc hodí (jazykově závislá) pravidla, někdy zas slovník, např. na interpretaci anglického členu "the" se hodí hlavně pravidla, ale bez slovníku to také nejde (srov. "the oak" proti "the one").

Pravidla můžu taky aplikovat buď jedním průchodem, nebo opakovaně. Typické druhy pravidel jsou následující:

1. morfosyntaktická pravidla – jedná se hlavně o určování slovních druhů apod. Používá se přitom hlavně slovník a statistické četnosti naměřené v nějakém korpusu. Příkladem takových pravidel může být i doplňování samohlásek v textech psaných souhláskovým písmem.
2. kontextová pravidla – Tato pravidla např. rozvíjejí zkratky, přibližují text čtené podobě.
3. strukturální pravidla – Výstup těchto pravidel se používá pro modelování prozódie – jde např. o identifikaci druhů vět, což umožní jejich správnou intonaci.
4. pravidla fonetického přepisu (letter-to-sound) – Tady se přímo převádí pravopis na výslovnost, mohou se používat různá pravidla pro výjimky (např. česky "diagram" změním na "dyagram", abych se vyhnul měkčení).

Modelování prozódie

Prozódie je vlastně ovlivňovaná "syntaxí", případně nějakými emocemi, jednotlivostmi mluvčího, ale ty se vystihnout nedají. Mělo by se dávat pozor i na mikroprozódii – tedy vystihnout prozodické fenomény, ale nenechat se zmást mikroprozodickými.

Intonace během řeči odpovídá změnám základní hlasové frekvence (F_0), na ostatních prozodických veličinách je víceméně nezávislá. Pro modelování intonace je nejznámější Fudžisakiho model. Ten sestává z phrase commands a accent commands. První typ pravidel je "trvanlivější", působí v podstatě na celou větu, vždy od daného času a s danou amplitudou (zvednutím nebo snížením F_0) a postupně doznívá. Druhý typ má kratší trvání, má definovaný čas začátku i konce a zase amplitudu. Výsledná F_0 v daném časovém bodě se (v logaritmické podobě) dá vyjádřit jako nějaká suma všech commandů, které působí, plus základní frekvence.

Prozodické modely je ale nutné nejprve natrénovat. Potřebuji tedy prozodický korpus, automatické nástroje na zpracování a prozodický model. Postup vytváření prozodického inventáře vypadá pak následovně:

$$\text{korpus} \rightarrow \text{detekce } F_0 \rightarrow \text{model} \rightarrow \underset{\text{rule extraction}}{\text{trénování /}} \rightarrow \text{pravidla (inventory)}$$

Krok trénování, extrakce pravidel probíhá buď automaticky za pomoci neuronové sítě (trénování), nebo ručně.

Výsledkem procesu by měl být prosodic inventory, tedy sada pravidel, jak upravovat prozodicky signál ve výstupu z TTS. Je to většinou malá množina nějakých hodnot – třeba informací o neuronové síti.

Syntéza řeči

Pro generování řeči ze zápisu hlásek se používá nějaký zjednodušený popis artikulace, podložený jistými předpoklady, tzv. řečový model. Pro syntézu existují dva hlavní druhy – buď copy synthesis, konkatenační syntéza, tedy syntéza na základě kopírování a slepování částí řečového inventáře, nebo rule-based synthesis, formant synthesis, syntéza založená na vytváření složeného zvuku za pomoci (frekvenčních) pravidel.

Pravidlová syntéza se používá většinou jenom v akademickém prostředí, až na pomůcky pro hyperrychlé čtení e-mailů. Projev většinou není příliš přirozený. Předpokládáme tu matematický model zjednodušeného artikulačního ústrojí a pravidla, popisující jeho změny. Ta pak zahrnují formanty samohlásek, transienty konsonantů, přítomnost základního tónu apod., všechno je v pravidlech relativně přímočaře. Model se vytváří ručně podle nějakého korpusu.

Kvalitu syntézy založená na kopírování určuje hlavně kvalita nahrávek v řečovém korpusu a také jejich reprezentativita. Korpus můžu získat dvěma způsoby – buď nahrávat televizní pořady, nebo výběrem vět, které někdo potom do korpusu přečte. Druhým způsobem můžu lépe pokrýt inventář cílového jazyka.

Chci mít výsledný korpus malý, aby ho mluvčí mohl přečíst najednou a bez změny podmínek (např. únavy hlasu). Postup je potom následující:

1. identifikace hlásek – Vyberu si, které hlásky potřebuji pro reprezentaci řeči v daném jazyce.
2. identifikace fonotaktiky – Zjistím, které kombinace vybraných hlásek se v jazyce vůbec můžou vyskytovat, tím zmenším počet kombinací.
3. kompozice korpusu – Ze všech možných kombinací hlásek, nalezených v předchozím kroku, složím psanou verzi korpusu.
4. nahrávání korpusu – Mluvčí přečte všechny věty, vložené do korpusu. Přitom by měl používat monotónní prozódii. Po nahrávání se vzorky normalizují na stejnou F_0.
5. vytvoření řečového inventáře – Protože pro každou kombinaci hlásek nepotřebuji více verzí, srovnám všechny dostupné a např. podle toho, jak moc se jejich F_0 blížila průměru, si vyberu tu nejlepší.

Pro výstup syntézy se nikdy nepoužívají samostatné hlásky, ale vždy kombinace dvou, tří nebo více hlásek, dvojhlásky apod. Projevuje se tu totiž důležitost koartikulace, navíc konsonanty jen "parazitují" na vokálech, samy stát nemohou, tedy samotné je extrahovat ani nemůžu. Pro konkatenaci zvuků potřebuji hlásky "stabilní", navíc vždy je potřeba nějaké vyhlazování zvuku. Tradičně se v konkatenační syntéze používají tzv. diphones, dvojzvuky – druhá polovina první, první polovina druhé hlásky, někde i delší úseky.

U složitějších systémů konkatenační syntézy nemám v řečovém inventáři pro každý diphone nebo triphone jen jednu zvukovou podobu, ale vybírám si z několika možností pomocí tzv. unit selection algorithm tu nejlepší pro dané místo v řeči. Přitom se zohledňuje prozódie, diskvalifikují se chyby výslovnosti apod., někdy se tak mohou použít i části slov úplně vcelku (na základě výběru). Pro výběr se používá upravený Viterbiho algoritmus (místo pravděpodobností přechodu uvažuju nějaké badness založené na podobné F_0, chybách výslovnosti, intenzitě a prozódi apod.).

Druhy Text-To-Speech Systémů

Time-Domain Pitch-Synchronous Overlap Add (TD-PSOLA)

Tento systém je příkladem konkatenativní syntézy, jde vlastně o velmi jednoduchý případ (dnes už relativně zastralý, používaný hlavně v 90. letech). Spočívá v tom, že každá hláska (jednotka řeči) je rozdělena na framy, krátké zvukové úseky během kterých se nemění F_0. V každém framu lze pozorovat pitch-periody, tedy jednotlivé kmity hlasu. Ty dávají možnost, jak měnit F_0 bez ohledu na kvalitu zvuku.

Mohu totiž jednotlivé framy skládat přes sebe a natahovat, pokud je upravím pomocí tzv. windowing funkce (funkce, která zesílí jen jednu pitch-periodu a postupně signál zeslabuje v jejím okolí až do ticha). Po použití windowing funkce na každou pitch-periodu pak výsledky můžu sečíst přes sebe i s nějakým posunutím. Tím dostanu signál, který může mít jinou F_0, ale jen neznatelně změněné vyšší frekvence (např. formanty). Někdy pitch-periody nesedí úplně přesně, ale díky windowing-funkci dojde k vyhlazení. Vyhazováním nebo duplikací pitch-period můžu (do určité míry) zvuk zrychlovat nebo zpomalovat.

Princip TD-PSOLA vypadá sice jednoduše, nutnou podmínkou jeho použití je ale spolehlivý detektor hlasové frekvence, jinak dochází k chybě fáze, phase mismatch hlasu (pitch-periody se netrefí přesně doprostřed kmitů). Na hranicích jednotlivých diphonů může dojít i k chybě spektra, spectral mismatch. Mám-li totiž dvě poloviny stejné samohlásky, které se trochu liší pod vlivem okolí, nedají se slepit úplně přesvědčivě. Poslední chybou, která se může v TD-PSOLA objevit, je chyba výšky hlasu, pitch mismatch. K té dojde, pokud dva přiléhající segmenty mají příliš odlišné F_0 (nesedí přesně na sebe).

Linear Prediction Coder (LPC) Speech Synthetizer

LPC syntéza vychází z modelu artikulačního ústrojí. Je to relativně stará technika, její výsledek ale nevypadá příliš přesvědčivě. Implementace v hardwaru ale není složitá, zvuk je srozumitelný i s minimálním inventářem.

Hlasové ústrojí si totiž lze představit jako na jedné straně otevřenou rezonační trubici (tube), ve které je na uzavřené straně zdroj zvuku (buzzer), který vytváří periodický signál. Když se nemění parametry tube ani buzzeru, pak vychýlení výsledné zvukové vlny v každém okamžiku (podle potřeb vzorkovací frekvence, kromě několika počátečních vzorků) se dá predikovat z určitého počtu předchozích vzorků.

Pro syntézu tedy vezmu řečový inventář a každou potřebnou jednotku rozdělím na framy, tedy časové úseky, kde jsou změny artikulace minimální. Pro každý frame potom odhadnu několik počátečních (např. 8) samplů, aby predikce vycházela s co nejmenší chybou. Tyto počáteční parametry se nazývají LPC coefficients. Odhad se typicky provádí metodou nejmenších čtverců.

Modelování pak provádím separátním ovládáním vlastností trubice i zdroje zvuku. Naměřené koeficienty nepoužívám pro generování zvuku přímo, protože mezi segmenty by vznikaly ostré předěly – zvuk se předem ještě vyhlazuje.

Problém je se simulací nosových hlásek, protože na to aproximace artikulačního ústrojí prostou trubicí nefunguje. Pokud bych chtěl trubici po části délky rozdělit, budu mít problém s nalezením počátečních LPC koeficientů.

Podobná technika (LPC komprese) se používá i v mobilních telefonech, protože aproximace parametrů je de facto druh ztrátové komprese.