Munich AUtomatic Segmentation (MAUS)
Phonemic Segmentation and Labeling
using the MAUS Technique

F. Schiel, Chr. Draxler, J. Harrington

Bavarian Archive for Speech Signals
Institute of Phonetics and Speech Processing
Ludwig-Maximilians-Universität München, Germany

www.bas.uni-muenchen.de
info@bas.uni-muenchen.de
Overview

- Statistical Segmentation and Labeling
- Super Short Introduction to MAUS
- Pronunciation Model: Building the Automaton
- Pronunciation Model: From Automaton to Markov Model
- Evaluation of Segmentation and Labeling
- Software Package MAUS
Let $\Psi$ be all possible Segmentation & Labeling (S&L) for a given utterance. Then the search for best S&L $\hat{K}$ is:

$$\hat{K} = \arg\max_{K \in \Psi} P(K|o) = \arg\max_{K \in \Psi} \frac{P(K)p(o|K)}{p(o)}$$

with $o$ the acoustic observation of the signal. Since $p(o) = \text{const}$ for all $K$ this simplifies to:

$$\hat{K} = \arg\max_{K \in \Psi} P(K)p(o|K)$$

with:

- $P(K)$ = apriori probability for a label sequence,
- $p(o|K)$ = the acoustical probability of $o$ given $K$ (often modeled by a concatenation of HMMs)
S&L approaches differ in creating $\Psi$ and modeling $P(K)$

For example: \textit{forced alignment}

$$||\Psi|| = 1 \quad \text{and} \quad P(K) = 1$$

hence only $p(o|K)$ is maximized.

Other ways to model $\Psi$ and $P(K)$:

- phonological rules resulting in $M$ variants with $P(K) = \frac{1}{M}$
- phonotactic n-grams
- lexicon of pronunciation variants
- \textbf{Markov process} (MAUS)
Short Introduction to MAUS

Diagram:
- Utterance
  - TEXT
  - PHONOLEX
    - Lexicon Lookup
  - Generator
    - Viterbi
      - Refinement
  - REFRUL
  - PHONRUL
  - MAUS Output
Short Introduction to MAUS
Building the Automaton

Start with the orthographic transcript:

\textit{heute Abend}

By applying lexicon-lookup and/or a test-to-phoneme algorithm produce a (more or less standardized) citation form in SAM-PA:

\texttt{hOYt@ ?a:b@nt}

Add word boundary symbols \#, form a linear automaton $G_c$:
Extend automaton $G_c$ by applying a set of substitution rules $q_k$ where each $q_k = (a, b, l, r)$ with
- $a$: pattern string
- $b$: replacement string
- $l$: left context string
- $r$: right context string

For example the rules
- $(/@n/, /m/, /b/, /t)$ and $(/b@n/, /m/, /a:, /t/)$

generate the reduced/assimilated pronunciation forms
- $/?a:bmt/$ and $/?a:mt/$

from the canonical pronunciation
- $/?a:b@nt/ \text{ (evening)}$
Building the Automaton

Applying the two rules to $G_c$ results in the automaton:
From Automaton to Markov Process

Add transition probabilities to the arcs of $G(N, A)$

- Case 1: all paths through $G(N, A)$ are of equal probability
  - Not trivial since paths can have different lengths!
  - Transition probability from node $d_i$ to node $d_j$:

$$P(d_j | d_i) = \frac{P(d_j)N(d_i)}{P(d_i)N(d_j)}$$

$N(d_i)$: number of paths ending in node $d_i$
$P(d_i)$: probability that node $d_i$ is part of a path

$N(d_i)$ and $P(d_i)$ can be calculated recursively through $G(N, A)$ (see Kipp, 1998 for details).
Example:
Markov process with 4 possible paths of different length

Total probabilities:

\[
\begin{align*}
1 \cdot \frac{3}{4} \cdot \frac{1}{3} \cdot 1 \cdot 1 &= \frac{1}{4} \\
1 \cdot \frac{1}{4} \cdot 1 \cdot 1 &= \frac{1}{4} \\
1 \cdot \frac{3}{4} \cdot \frac{1}{3} \cdot 1 &= \frac{1}{4} \\
1 \cdot \frac{3}{4} \cdot \frac{1}{4} \cdot 1 \cdot 1 &= \frac{1}{4}
\end{align*}
\]
Case 2: all paths through $G(N, A)$ have a probability according to the individual rule probabilities along the path through $G(N, A)$

Again not trivial, since contexts of different rule applications may overlap!
This may cause total branching probabilities $> 1$

*Please refer to Kipp, 1998 for details to calculate correct transition probabilities.*
From Markov Process to Hidden Markov Model

True HMM: add emission probabilities to nodes $N$ of $G_c$.

$\rightarrow$ Replace the phonemic symbols in $N$ by mono-phone HMM.

The search lattice for previous example:
Word boundary nodes ‘#’ are replaced by a optional silence model:

Possible silence intervals between words can be modeled.
How to evaluate a S&L system?

Required: reference corpus with hand-crafted S&L (‘gold standard’).

Usually two steps:

1. Evaluate the accuracy of the label sequence (transcript)
2. Evaluate the accuracy of segment boundaries
Evaluation of Label Sequence

Often used for label sequence evaluation: Cohen’s $\kappa$

$\kappa = \text{amount of overlap between two transcripts (system vs. gold standard); independent of the symbol set size (Cohen 1960).}$

We consider $\kappa$ not appropriate for S&L evaluation, since

- no gold standard exists in phonemic S&L
- different symbol set sizes do not matter in S&L
- the task difficulty is not considered (e.g. read vs. spontaneous speech)
Proposal: *Relative Symmetric Accuracy (RSA)* =

\[
RSA = \frac{\widehat{SA}_{hs}}{\widehat{SA}_{hh}} \times 100\%
\]

= the ratio from average symmetric system-to-labeler agreement $\widehat{SA}_{hs}$ to average inter-labeler agreement $\widehat{SA}_{hh}$. 
Evaluation of Label Sequence

German MAUS:
- 3 human labelers
- spontaneous speech (Verbmobil)
- 9587 phonemic segments

Average system - labeler agreement
Average inter - labeler agreement
Relative symmetric accuracy

\[ \hat{SA}_{hs} = 81.85\% \]
\[ \hat{SA}_{hh} = 84.01\% \]
\[ RSA = 97.43\% \]
Evaluation of Segmentation

- No standardized methodology
- Problem: insertions and deletions
- Solution: compare only matching segments
- Often: count boundary deviations greater than threshold (e.g. 20msec) as errors
- Better: deviation histogram measured against all human segmenters
Evaluation of Segmentation

German MAUS:

Note: center shift typical for HMM alignment
MAUS software package:

ftp://ftp.bas.uni-muenchen.de/pub/BAS/SOFTW/MAUS

MAUS requires

- UNIX System V or cygwin
- Gnu C compiler
- HTK (University of Cambridge)

Current language support: German, English, Hungarian, Icelandic, Estonian, Portuguese, Spanish

A MAUS web services is currently in alpha.

*If interested in a demo, please contact me after the talk.*
References

- MAUS: ftp://ftp.bas.uni-muenchen.de/pub/BAS/SOFTW/MAUS
- CLARIN: http://www.clarin.eu/
Software Package MAUS

How to adapt MAUS to a new language?

Several possible ways (in ascending performance and effort):

- Define a mapping from the phoneme set of the new language to the German set (or any other available language in MAUS). Constrain pronunciation to canonical form.
  
  **Effort:** nil
  
  **Performance:** for some languages surprisingly good.
Software Package MAUS

- Hand craft pronunciation rules (depending on language not more than 10-20) and run MAUS in the ’manual rule set’ mode.
  *Effort:* small
  *Performance:* Very much dependent of the language, the type of speech, the speakers etc.

- Adapt HMM to a corpus of the new language using an iterative training schema (script `maus.iter`). Corpus does not need to be annotated.
  *Effort:* moderate (if corpus is available)
  *Performance:* For most languages very good, depending on the adaptation corpus (size, quality, match to target language etc.)
Software Package MAUS

- Retrieve statistically weighted pronunciation rules from a corpus. The corpus needs to be at least of 1 hour length and segmented/labeled manually.

  *Effort:* high.
  
  *Performance:* Unknown.