Confidence Measures for Large Vocabulary Continuous Speech Recognition

Present by Tzan-Hwei Chen


Introduction (1/3)

• It is extremely important to able to make an appropriate and reliable judgement based on the error-prone ASR result.

• researchers have proposed to compute a score (preferably 0~1), called confidence measure (CM) to indicate reliability of any recognition decision made by ASR system.

• Under a different name, namely *utterance verification*, Rose et al.(1995) first formally cast the CM problem in speech recognition as a statistical hypothesis testing problem
Introduction (2/3)

• First of all, we can backtrack some early research on CM to rejection in word-spotting systems.

• Other early CM-related works lie in automatic detection of new words in LVCSR.

• From the past few years, the CM is applied to more and more research areas, ex:
  – To improve speech recognition
  – The algorithm about look-head in LVCSR
  – Guide the system to perform unsupervised learning
  – ...

Introduction (3/3)

• all methods proposed for computing CMs can be roughly classified into three major categories:

  – Predictor features.

  – Posterior probability

  – Utterance verification (UV)
How to compute CMs (1/16)

• Predictor features:
  – Collected during decoding procedure and may include acoustic as well as language information
  – Any feature can be called a predictor if its p.d.f of correctly recognized words is clear distinct form that of misrecognized words

![Feature value graph]

Feature value
How to compute CMs (2/16)

• Predictor features (cont) : Some common predictor features
  – Pure normalized likelihood score related : acoustic score per frame.
  – N-best related : count in the N-best list, N-best homogeneity score, …
  – Acoustic stability :

Different LM
How to compute CMs (3/16)

• Predictor features (cont) : Some common predictor features
  
  – Hypothesis density :

  \[
  D(t') = \left\{ a' : [w_{a'}, s_{a'}, e_{a'}] \in WG \land s_{a'} \leq t' \leq e_{a'} \right\}
  \]

  \[
  HD(a : [w_{a}, s_{a}, e_{a}]) = \frac{1}{e_{a} - s_{a} + 1} \sum_{t=s_{a}}^{e_{a}} D(t)
  \]

  – Posterior probability

  – Log-likelihood-ratio

  – Duration related : phone duration, word duration

  – LM score
How to compute CMs (4/16)

- Predictor features (cont) : Some common predictor features
  - Parsing related:

```
!IS!
ROOT
  ACTION
  SEGMENT
    LOC-FR
      fb
    DATE-DEP
      null
      null
      null
      null
      null
      null
      null
      loc-fr
      null
      null
      date-dep
      .
```

<table>
<thead>
<tr>
<th>Parse Tree Probabilities</th>
<th>!IS!</th>
<th>[ROOT]</th>
<th>:NONE_fb</th>
<th>i_null</th>
<th>want_null</th>
<th>to_null</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node Extension</td>
<td>1</td>
<td>0.999516</td>
<td>0.984263</td>
<td>0.988049</td>
<td>0.994018</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.999522</td>
<td>0.984119</td>
<td>0.992231</td>
<td>0.995098</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node Extension</th>
<th>ACTION</th>
<th>SEGMENT</th>
<th>:fly_flights</th>
<th>[LOC-FR]</th>
<th>from_null</th>
<th>LOC_loc-fr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.997064</td>
<td>1</td>
<td>0.990373</td>
<td>0.998472</td>
<td>0.702338</td>
<td>0.991334</td>
</tr>
<tr>
<td></td>
<td>0.995984</td>
<td>0.995123</td>
<td>1</td>
<td>0.999872</td>
<td>0.999981</td>
<td>0.994569</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node Extension</th>
<th>LOC-FR</th>
<th>DATE-DEP</th>
<th>:on_null</th>
<th>DATE_date-dep</th>
<th>DATE-DEP</th>
<th>SEGMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>0.544135</td>
<td>0.998505</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0.998472</td>
<td>0.990492</td>
<td>0.974586</td>
<td>0.999671</td>
<td>1</td>
<td>0.995123</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node Extension</th>
<th>ACTION</th>
<th>..</th>
<th>ROOT</th>
<th>!IS!</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.997064</td>
<td>1</td>
<td>0.999516</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0.995984</td>
<td>0.999519</td>
<td>0.999522</td>
<td>1</td>
</tr>
</tbody>
</table>

```
How to compute CMs (5/16)

• Predictor features (cont) : some people attempt to combine several different features for a better performance
  – Line discriminant function
  – Generalized linear model
  – Neural networks
  – Decision tree
  – Support vector machine
  – Boosting
  – Naïve bayes classifier
How to compute CMs (6/16)

• Posterior probability:
  – The conventional ASR system

\[ \hat{W} = \arg \max_{W \in \Sigma} P(W \mid X) \]

\[ = \arg \max_{W \in \Sigma} \frac{P(X \mid W)P(W)}{P(X)} \text{ ignore} \]

\[ = \arg \max_{W \in \Sigma} P(X \mid W)P(W) \]

– In theory, we should compute \( P(X) \) as follow:

\[ P(X) = \sum_{H} P(X, H) \]

Impossible to estimate in a precise manner
How to compute CMs (7/16)

- Posterior probability (cont):
  - Word graph is a compact and fairly accurate representation of all alternative competing hypotheses
How to compute CMs (8/16)

• Posterior probability (cont):  
  – The any arc in word graph can be computed as:

\[
p([w, s, t] | x^T_1) = \frac{1}{p(x^T_1)} \sum_w \sum_{w'} p(x^{s-1}_1 | W) p(x^t_s | w) \\
\quad \cdot p(x^T_{t+1} | W) \cdot p(WwW')
\]

  – some issue are addressed and the word posterior probability is generalized
    • Reduced search space
    • Relaxed time registration
    • Optimal acoustic and language model weights
How to compute CMs (9/16)

• Posterior probability (cont):
  – some approaches

\[
p([w; s, e] \mid X) = \frac{\sum_{M, \forall n, 1 \leq n \leq M \atop w_n = w} \prod_{m=1}^{M} p^{\alpha}(x_{s_m}^{e_m} \mid w_m) p^{\beta}(w_m \mid w_1^M)}{p(X)}
\]  

(2)

\[
C_{\text{sec}} ([w, s, e]) = \sum_{[w]_n^{e_n} : (s, e) \cap (s', e') \neq \phi} p([w, s', e'] \mid X)
\]  

(3)

\[
C_{\text{med}} ([w, s, e]) = \sum_{[w]_n^{e_n} : s' \leq s + \frac{e - s}{2} \leq e'} p([w, s', e'] \mid X)
\]  

(4)

\[
C_{\text{max}} ([w, s, e]) = \max_{t \in \{s \ldots e\}} \sum_{[w]_n^{e_n} : s' \leq t \leq e'} p([w, s', e'] \mid X)
\]  

(5)
How to compute CMs (10/16)

• Posterior probability (cont):
  – The drawbacks of above methods – needed the additional pass.
  – So in [9], they proposed the “local word confidence measure”

\[
C([w, s, e]) = \frac{\max(p(x^e | w))^\alpha p(w)^\beta}{\sum \max(p(x^{e'} | w'))^\alpha p(w')^\beta}
\]

(6)

\[
C([w, s, e]) = \frac{\max(p(x^e | w))^\alpha \sum w' p(w | w'_h)^\beta}{\sum \max(p(x^{e'} | w'))^\alpha \sum w'_h p(w'_h | w'_h)^\beta}
\]

(7)

\[
C([w, s, w]) = \frac{\max(p(x^e | w))^\alpha \sum w_h \sum w_f \{p(w | w_h)p(w_f | w)_i^\beta}{\sum \max(p(x^{e'} | w'))^\alpha \sum w_h \sum w_f \{p(w' | w_h')p(w'_f | w'_f)_j^\beta}
\]

(8)
How to compute CMs (11/16)

• Posterior probability (cont):
  – Impact of Word Graph Density on the Quality of Posterior Probability

<table>
<thead>
<tr>
<th></th>
<th>Verbmobil</th>
<th></th>
<th>NaDia</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>WGD</td>
<td>CER [%]</td>
<td>WGD</td>
<td>CER [%]</td>
<td></td>
</tr>
<tr>
<td>17.7</td>
<td>19.7</td>
<td>30.9</td>
<td>10.9</td>
<td></td>
</tr>
<tr>
<td>206.5</td>
<td>16.7</td>
<td>383.5</td>
<td>10.2</td>
<td></td>
</tr>
<tr>
<td>736</td>
<td>16</td>
<td>1170.4</td>
<td>10.3</td>
<td></td>
</tr>
</tbody>
</table>

Baseline 27.3 15.4

\[ CER = \frac{\text{the number of classification errors}}{\text{the total number of recognized words}} \]

\[ \text{baseline} = \frac{\text{sub. + ins.}}{\text{the total number of recognized words}} \]
How to compute CMs (12/16)

• Utterance verification
  – The CM problem is formulated as a statistical hypothesis testing problem.
  
  – Under the framework of UV, we propose two complementary hypotheses

  \[ H_0 \] (Null Hypothesis): \( X \) is correctly recognized and truly comes from model \( \lambda_w \)
  
  \[ H_1 \] (Alternative Hypothesis): \( X \) is wrongly recognized and is NOT from model \( \lambda_w \)

  – Then we test \( H_0 \) against \( H_1 \)

  \[
  \text{likelihood ratio testing (LRT)} = \frac{P(X \mid H_0)}{P(X \mid H_1)} \overset{H_0}{\underset{H_1}{\tau}}
  \] (9)
How to compute CMs (13/16)

• Utterance verification (cont)
  – As pointed out by Lee(2001), the above LRT score can be transformed to a CM based on a monotonic 1-1 mapping function.

  – The major difficulty with LRT is how to model the alternative hypothesis.

  – In practice, the same HMM structure is adopted to model the alternative hypothesis.

  – A discriminative training procedure plays a crucial role in improving modeling performance.
How to compute CMs (14/16)

• Utterance verification (cont) – MCE training

frame based distance:

\[ r_{ij}(y_t) = \log(a_{ij}^c b_j^c(y_t)) - \log(a_{ij}^a b_j^a(y_t)) \]  \hspace{1cm} (10)

segment based distance is obtained by averaging the frame based distances as

\[ R_u(Y^u) = \frac{1}{t_{f_u} - t_{i_u} + 1} \sum_{t=t_{i_u}}^{t_{f_u}} r_{q_{t-1}q_t}(y_t) \]  \hspace{1cm} (11)

where the indicator function \( \delta(u) \) is defined as

\[ \delta(u) = \begin{cases} -1, & u \in \text{correct} \\ 1, & u \in \text{imposter} \end{cases} \]

A gradient update is performed on the expected cost \( E\{F_u(Y^u, \Lambda^u)\} \)

\[ \Lambda_{n+1}^u = \Lambda_n^u - \epsilon \nabla E\{F_u(Y^u, \Lambda^u)\} \]  \hspace{1cm} (12)
How to compute CMs (15/16)

- Incorporation of high-level information for CM
  - LSA:

\[
\begin{align*}
\begin{bmatrix}
  s_1 & s_2 & \ldots & s_n \\
  w_1 & w_2 & \ldots & w_m
\end{bmatrix}
  &=
  \begin{bmatrix}
    \mathbf{U} \\
    \Sigma \\
    \mathbf{V}^T
  \end{bmatrix}
\end{align*}
\]

- The key property of LSA is that words whose vectors are “close” correspond to semantically similar words.

- These similarities can be used to provide an estimate of the likelihood of the words co-occurring within the same utterance.
How to compute CMs (16/16)

• Incorporation of high-level information for CM (cont)
  – Inter-word mutual information:

  Assume \( N(x, y) \) begin the co-occurrence times of word \( x \) and word \( y \) in all training documents, the joint probability \( P(x, y) \) is:

  \[
  P(x, y) = \frac{N(x, y)}{\sum_{x,y} N(x, y)}
  \]

  Mutual information between any two words \( x \) and \( y \) can be calculated as follows

  \[
  MI = \log\left(\frac{P(x, y)}{P(x)P(y)}\right)
  \]

  CM of each recognized word is calculated as the average mutual information of this word with all other recognized words.
Some recent applications (1/9)

• Verification-Based Fast-Match
  – Introduction

![Diagram](Image)
Some recent applications (2/9)

- Verification-Based Fast-Match (cont)
  - For a fast-match the null and alternative hypothesis testing for phoneme $\alpha$ can be written as:

  \[ H_0: \alpha \text{ starts at time } t \]
  \[ H_1: \alpha \text{ does not start at time } t \]

  \[
  LRT = \frac{P(X \mid H_0)}{P(X \mid H_1)} \begin{array}{c} H_0 \\ \downarrow \\ \tau \\ H_1 \end{array} \]

  \[ P(X \mid H_0) \equiv P(x_i^{t+d_\alpha} \mid \alpha) \]
  \[ P(X \mid H_1) \equiv P(x_i^{t+d_\alpha} \mid \bar{\alpha}) \]
Some recent applications (3/9)

- Word error minimization:
  - Statistical decision theory aims at minimizing the expected of making error
    \[ w_{1}^{N*} = \arg \max_{w_{1}^{N}} P(w_{1}^{N} | x_{1}^{T}) \]  
    \[ (13) \]
  - To assume the boundary time of word sequence is given [5]:
    \[ p(w_{1}^{N} | x_{1}^{T}) = p([w, s, t]_{1}^{N} | x_{1}^{T}) \]
    \[ = \prod_{n=1}^{N} p([w_{n}, s_{n}, t_{n}] | [w, s, t]_{1}^{n-1}, x_{1}^{T}) \]
    \[ = \prod_{n=1}^{N} p([w_{n}, s_{n}, t_{n}] | x_{1}^{T}) \]  
    \[ (14) \]
Some recent applications (4/9)

• Word error minimization (cont):
  – Minimizing the expected SER does not necessarily minimizing the expected WER

  – In [7]

\[
W_1^N = \arg \min_{w_i^N} \text{WER}(w_i^N \mid x_1^T) \quad (15)
\]

\[
\text{WER}(w_i^N \mid x_1^T) = 1.0 - \frac{1}{N} \sum_i \left \{ P(w_i = \text{correct}) \times P(w_i \mid x_1^T) \right \} \quad (16)
\]
Some recent applications (5/9)

- Word error minimization (cont):
  - In this very general framework
    \[
    w_{i}^{N*} = \arg \min_{w_{i}^{N}} \left\{ \sum_{v_{i}^{M}} C(w_{i}^{N}, v_{i}^{M}) \cdot p(v_{i}^{M} | x_{i}^{T}) \right\}
    \]
    \[\text{(17)}\]
  - The easiest way to overcome this mismatch it to use the same cost function – Levensthein distance
  - In (Stolcke et. al 1997), the pairwise alignment is restricted to N-best list.
  - Let us assume that sub. were the one type of error.
    - A dynamic programming alignment would thus not be necessary.
Some recent applications (6/9)

- Word error minimization (cont):
  - With these considerations and with the fact that a word graph contains the start and end time:

\[
C(w^N_1, v^M_1) = \sum_{n=1}^{N} \sum_{t=s_n}^{t=e_n} \frac{1 - \delta(w_n, v_t)}{1 + \alpha(e_n - s_n - 1)}
\]  

(18)
Some recent applications (7/9)

- Word error minimization (cont):
  - Correlation analysis
Some recent applications (8/9)

- **Word error minimization (cont):**

  Time Frame Error decoding [6]

  \[
  \{[w,s,e]_i^N\}_{opt} = \arg \min_{[w,s,e]_i^N} \left\{ \sum_{[v,s',e']_i^M} \ell([w,s,e]_i^N,[v,s',e']_i^M)P([v,s',e']_i^M | x_i^T) \right\}
  \]

  \[
  = \arg \min_{[w,s,e]_i^N} \left\{ \sum_{[v,s',e']_i^M} \sum_{n=1}^N \frac{\sum_{t=s_n}^{e_n} 1 - \delta(w_n,v_t)}{1 + \alpha(e_n - s_n - 1)} P([v,s',e']_i^M | x_i^T) \right\}
  \]

  \[
  = \arg \min_{[w,s,e]_i^N} \left\{ \sum_{n=1}^N \sum_{t=s_n}^{e_n} \frac{P([v,s',e']_i^M | x_i^T) - \sum_{t=s_n}^{e_n} \delta(w_n,v_t)P([v,s',e']_i^M | x_i^T)}{1 + \alpha(e_n - s_n - 1)} \right\}
  \]

  \[
  = \arg \min_{[w,s,e]_i^N} \left\{ \sum_{n=1}^N \sum_{t=s_n}^{e_n} \frac{1 - \sum_{t=s_n}^{e_n} \delta(w_n,v_t)P([v,s',e']_i^M | x_i^T)}{1 + \alpha(e_n - s_n - 1)} \right\}
  \]
Some recent applications (9/9)

• Word error minimization (cont):

\[
\sum_{[v,s',e']^M_{[i]}} \delta(w_n, v_t) P([v, s', e']^M \mid x^T_t)
\]

\[
= \sum_{[v,s',e']^M_{[i]}} \sum_{v_m:s_m \leq t \leq e'_m} \delta(w_n, v_t) P([v, s', e']^M \mid x^T_t)
\]

\[
= \sum_{[v,s',e'], v : s' \leq t \leq e'} \delta(w_n, v) P([v, s', e'] \mid x^T_t)
\]

\[
= p(w_n \mid t, x^T_t)
\]

\[
\sum_{t=s_n}^{t=e_n} \left[ 1 - p(w_n \mid t, x^T_t) \right]
\]

\[
\frac{1 + \alpha(e_n - s_n - 1)}{1 + \alpha(e_n - s_n - 1)}
\]

Can be interpreted as the normalized Probability of a word being incorrect.
Summary

- Almost all CMs in acoustic level fundamentally rely almost entirely on a single information source.

- We believe it is critical to improve performance of CMs by taking this segmentation issue into account.