Confidence Measures for Large Vocabulary Continuous Speech Recognition

Present by Tzan-Hwei Chen

Reference (1/3)

- [1] H. Jiang ,“Confidence Measures for Speech Recognition : A Survey”, Speech communication 2005 .
- [2] F. Wessel , R. Schluter, K. Macherey, and H. Ney, “Confidenc Measures for Large Vocabulary Continuous Speech Recognition”, IEEE SAP 2001.
- [3] R. Sarikaya, Y. Gao, M. Picheny and H. Erdogan, “Semantic Confidence Measurement for Spoken Dialog Systems.”, IEEE SAP 2005.
- [4] M. Afify, F. Liu, H. Jiang and O. Siohan, “A New Verification-based Fast-Match for Large Vocabulary Continuous Speech Recognition” IEEE SAP 2005

Reference (2/3)

- [5] F. Wessel , R. Schluter and H. Ney, “Using Posterior Word Probabilities for Improved Speech Recognition”, ICASSP 2000.
- [6] F. Wessel , R. Schluter and H. Ney, “Explicit Word Error Minimization Using Word Hypothesis Posterior Probabilities”, ICASSP 2001.
- [7] A. Kobayashi, K. Onoe, S. Sato and T.Imai, “Word Error Minimization Using an Integrate Confidence Measure”, INTERSPEECH 2005.
- [8] F. K. Soong and W. K. Lo, “Generalized Posterior Probability for Minimum Error Verification of Recognized Sentences”, ICASSP 2005

Reference (3/3)

- [9]J. Razik, O. Mella, D. Fohr, J.-P Haton, “Local Word Confidence Measure Using Word Graph and N-Best List.”
- [10]T, Fabian, R. Lieb, G. Ruske, M. Thomae, “Impact of Word Graph Density on the Quality of Posterior Probability Based Confidence Measures.

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)

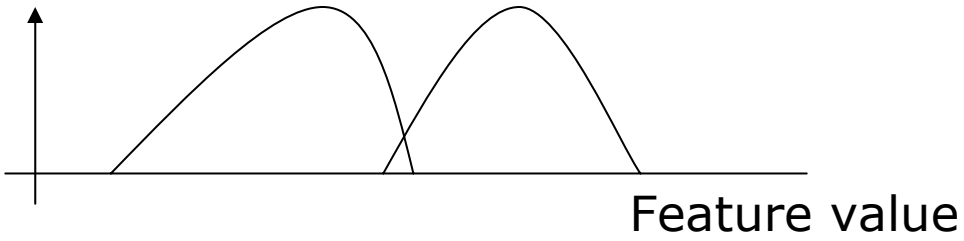
- First of all, we can backtrack some early research on CM to rejection in word-spotting systems.
- Other early CM-related works lie in automatic detection of new words in LVCSR.
- From the past few years, the CM is applied to more and more research areas, ex :
  - To improve speech recognition
  - The 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)

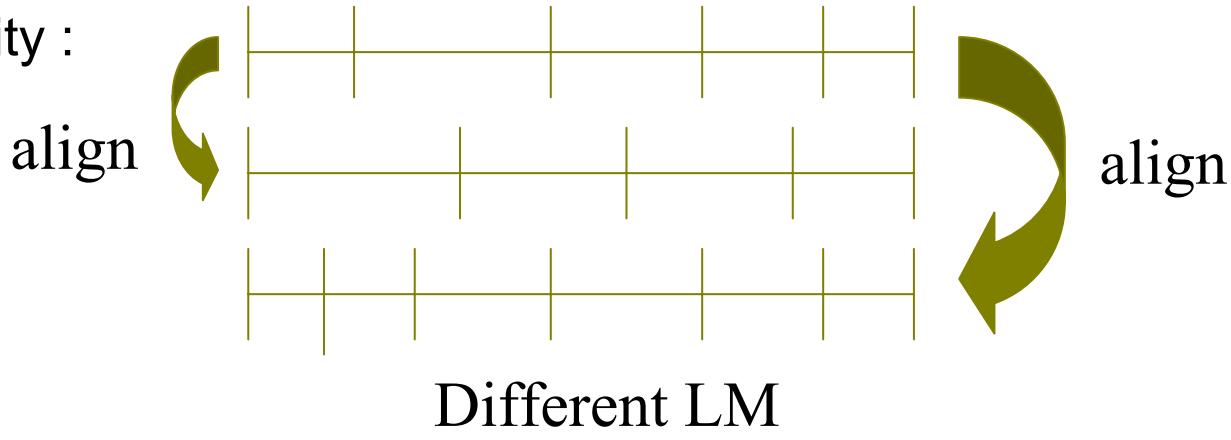
How to compute CMs (1/16)

- 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



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 :



How to compute CMs (3/16)

- Predictor features (cont) : Some common predictor features
  - Hypothesis density :

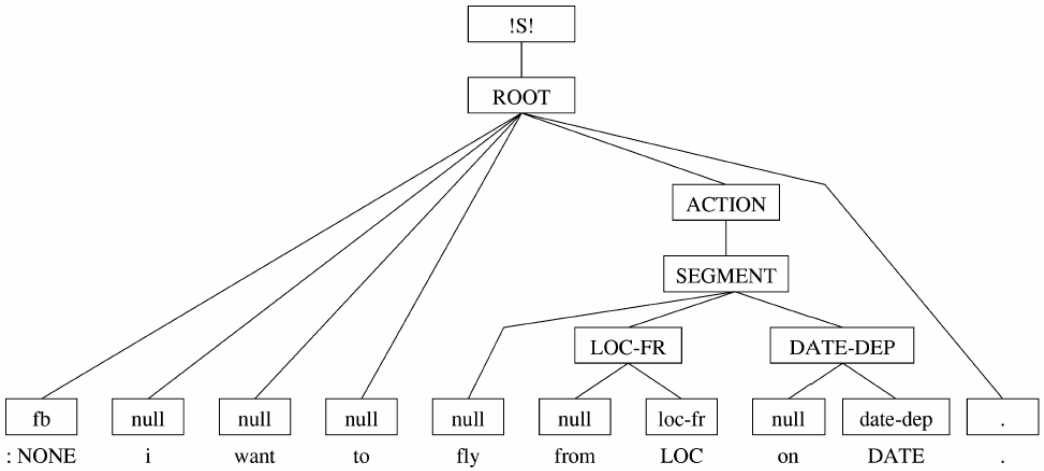
$$D(t') = \left|\left\{a' : [w_{a'}; s_{a'}, e_{a'}] \in WG \wedge s_{a'} \le t' \le e_{a'}\right\}\right|$$

$$HD(a : [w_a; s_a, e_a]) = \frac{1}{e_a - s_a + 1} \sum_{t=s_a}^{e_a} D(t)$$

  - 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 :



Parse Tree Probabilities						
	[!S!	[ROOT	:NONE_fb	i_null	want_null	to_null
Node	1	0.999516	1	0.984263	0.988049	0.994018
Extension	1	0.999522	0.984119	0.992231	0.995098	0.995098
	[ACTION	[SEGMENT	fly_flights	[LOC-FR	from_null	LOC_loc-fr]
Node	0.997064	1	0.990373	1	0.702338	0.991334
Extension	0.995984	0.995123	1	0.998472	0.999981	0.994569
	LOC-FR]	[DATE-DEP	on_null	DATE_date-dep	DATE-DEP]	SEGMENT]
Node	1	1	0.544135	0.998505	1	1
Extension	0.998472	0.990492	0.974586	0.999671	0.990492	0.995123
	ACTION]	...	ROOT]	!S!]		
Node	0.997064	1	0.999516	1		
Extension	0.995984	0.999519	0.999522	1		

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

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

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

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

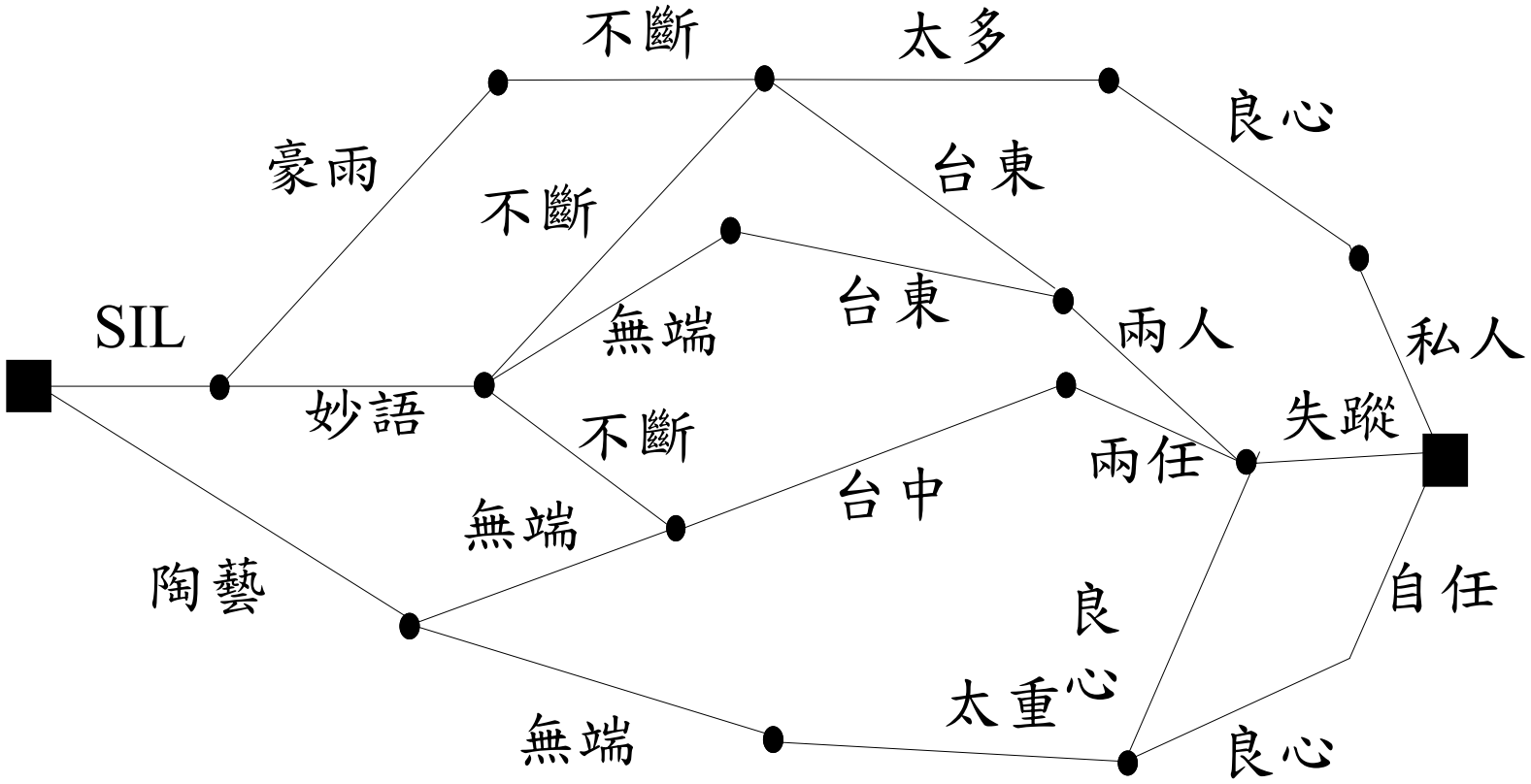
  - In theory, we should compute P(X) as follow :

$$P(X) = \sum_{H} P(X, H)$$

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 :

$$p([w,s,t]\,|\,x_1^T) = \frac{1}{p(x_1^T)} \sum_{W} \sum_{W'} p(x_1^{s-1}\,|\,W)\,p(x_s^t\,|\,w) \cdot p(x_{t+1}^T\,|\,W) \cdot p(WwW') \qquad (1)$$

  - 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

$$p([w;s,e]\,|\,X)=\sum_{\substack{M,[w;s,t]_1^M\\ \exists n,1\le n\le M\\ w_n=w\\ (s_n,t_n)\cap(s,t)\neq\phi}}\frac{\prod_{m=1}^{M}p^{\alpha}(x_{s_m}^{e_m}\,|\,w_m)p^{\beta}(w_m\,|\,w_1^M)}{p(X)} \qquad (2)$$

$$C_{\text{sec}}([w,s,e])=\sum_{\substack{[w]_{s'}^{e'}:\\ (s,e)\cap(s',e')\neq\phi}}p([w,s',e']\,|\,X) \qquad (3)$$

$$C_{med}([w,s,e])=\sum_{\substack{[w]_{s'}^{e'}:\\ s'\le[s+\frac{e-s}{2}]\le e'}}p([w,s',e']\,|\,X) \qquad (4)$$

$$C_{\max}([w,s,e])=\max_{t\in\{s\ldots e\}}\sum_{\substack{[w]_{s'}^{e'}:\\ s'\le t\le e'}}p([w,s',e']\,|\,X) \qquad (5)$$

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”

$$C([w,s,e]) = \frac{\max(p(x_s^e \mid w))^{\alpha}\, p(w)^{\beta}}{\sum_{[w',s',e']\in E} \max(p(x_{s'}^{e'} \mid w'))^{\alpha}\, p(w')^{\beta}} \qquad (6)$$

$$C([w,s,e]) = \frac{\max(p(x_s^e \mid w))^{\alpha} \sum_{w'} p(w \mid w_h)^{\beta}}{\sum_{[w',s',e']\in E} \max(p(x_{s'}^{s'} \mid w'))^{\alpha} \sum_{w'_h} p(w' \mid w'_h)^{\beta}} \qquad (7)$$

$$C([w,s,w]) = \frac{\max(p(x_s^e \mid w))^{\alpha} \sum_{w_h} \sum_{w_f} \{p(w \mid w_h)\, p(w_f \mid w)\}^{\beta}}{\sum_{[w',s',e']\in E} \max(p(x_{s'}^{e'} \mid w'))^{\alpha} \sum_{w_h} \sum_{w_f} \{p(w' \mid w'_h)\, p(w'_f \mid w')\}^{\beta}} \qquad (8)$$

How to compute CMs (11/16)

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

Verbmobil		NaDia	
WGD	CER [%]	WGD	CER [%]
17.7	19.7	30.9	10.9
206.5	16.7	383.5	10.2
736	16	1170.4	10.3
Baseline	27.3		15.4

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

$$baseline = \frac{\text{sub.} + \text{ins.}}{\text{the total number of recognized words}}$$

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 Hypothese) : X is correctly recognized and truly comes from model λ_W

    H_1 (Alternative Hypothese) : X is wrongly recognized and is NOT from model λ_W
  - Then we test H_0 against H_1

$$\text{likelihood ratio testing(LRT)} = \frac{P(X \mid H_0)}{P(X \mid H_1)} \underset{H_1}{\overset{H_0}{\gtrless}} \tau \qquad (9)$$

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:

$$r_{\mathrm{ij}}(y_t) = \log(a_{ij}^c b_j^c(y_t)) - \log(a_{ij}^a b_j^a(y_t)) \qquad (10)$$

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

$$R_u(Y^u) = \frac{1}{t_{f_u} - t_{i_u} + 1} \sum_{t=t_{i_u}}^{t=t_{f_u}} r_{\mathrm{q_{t-1}}q_t}(y_t) \qquad (11)$$

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

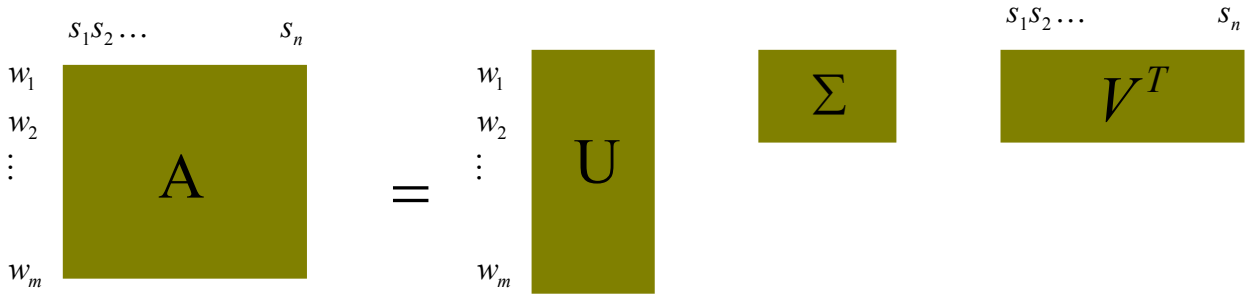
$$\delta(u) = \begin{cases} -1, & u \in \text{correct} \\ 1, & u \in \text{imposter} \end{cases}$$

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

$$\Lambda_{n+1}^u = \Lambda_n^{\mathrm{u}} - \varepsilon \nabla E\{F_u(Y^u, \Lambda^u)\} \qquad (12)$$

How to compute CMs (15/16)

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



  - 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 :

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

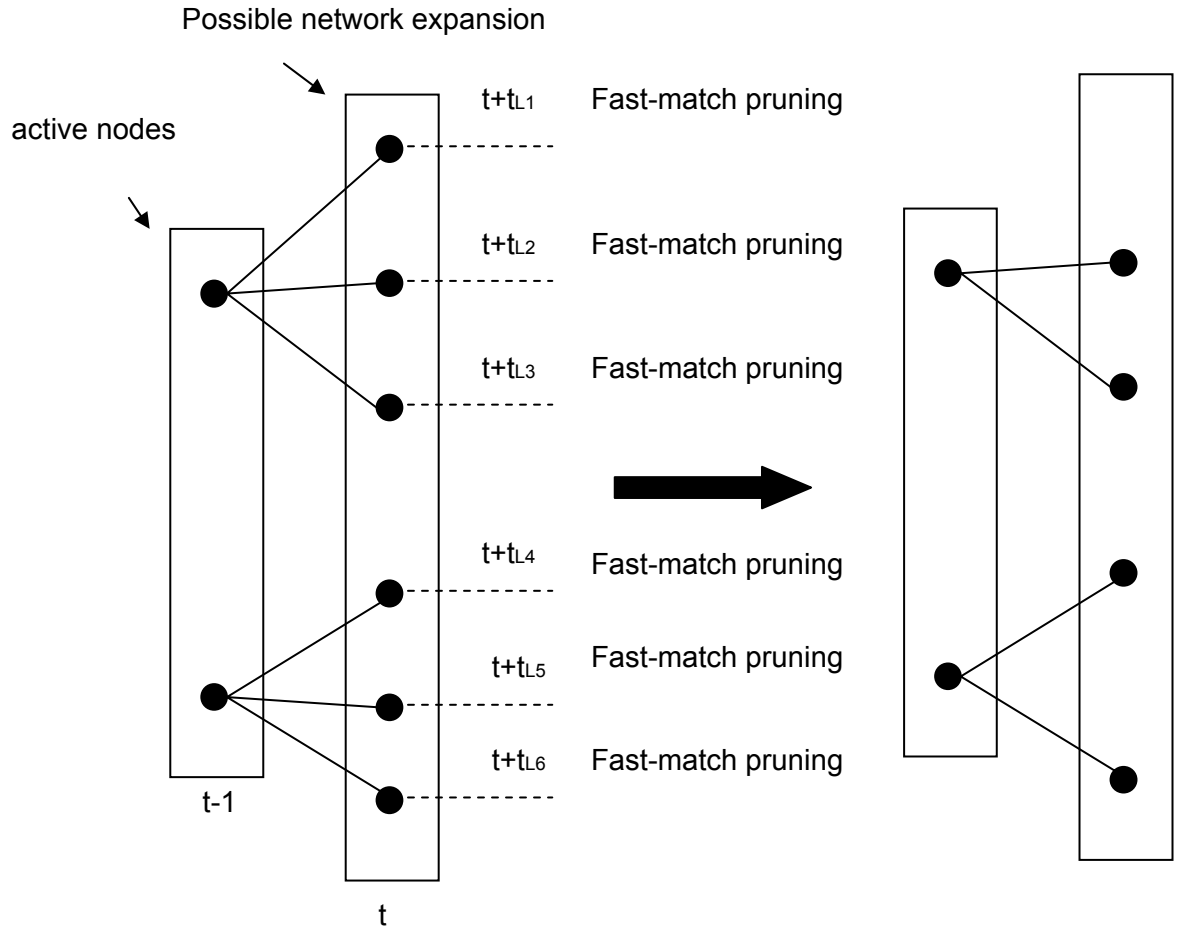
Mutual information between any two words x and y can be calculated as follows

$$MI = \log\left(\frac{p(x, y)}{p(x)p(y)}\right)$$

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



Some recent applications (2/9)

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

$H_0 : \alpha$ starts at time t

$H_1 : \alpha$ does not start at time t

$$LRT = \frac{P(X \mid H_0)}{P(X \mid H_1)} \underset{H_1}{\overset{H_0}{\gtrless}} \tau$$

$$P(X \mid H_0) \equiv P(x_t^{t+d_\alpha} \mid \alpha)$$

$$P(X \mid H_1) \equiv P(x_t^{t+d_\alpha} \mid \bar{\alpha})$$

Some recent applications (3/9)

- Word error minimization :
  - Statistical decision theory aims at minimizing the expected of making error

$$w_1^{N^*} = \arg\max_{w_1^N} P(w_1^N \mid x_1^T) \qquad (13)$$

  - To assume the boundary time of word sequence is given [5]:

$$p(w_1^N \mid x_1^T) = p([w,s,t]_1^N \mid x_1^T)$$

$$= \prod_{n=1}^{N} p([w_n,s_n,t_n] \mid [w,s,t]_1^{n-1}, x_1^T)$$

$$= \prod_{n=1}^{N} p([w_n,s_n,t_n] \mid x_1^T) \qquad (14)$$

Some recent applications (4/9)

- Word error minimization (cont) :
  - Minimizing the expected SER does not necessarily minimizing the expected WER
  - In [7]

$$w_1^{N^*} = \arg\min_{w_1^N} \mathrm{WER}(w_1^N \mid x_1^T) \qquad (15)$$

$$WER(w_1^N \mid x_1^T) = 1.0 - \frac{1}{N}\sum_i \left\{P(w_i = correct) \times P(w_i \mid x_1^T)\right\} \qquad (16)$$

Some recent applications (5/9)

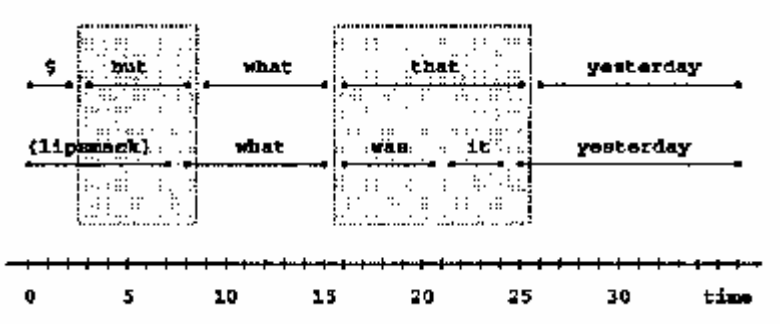
- Word error minimization (cont) :
  - In this very general framework

$$w_1^{N^*} = \arg\min_{w_1^N} \left\{ \sum_{v_1^M} C(w_1^N, v_1^M) \cdot p(v_1^M \mid x_1^T) \right\} \qquad (17)$$

  - 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 :

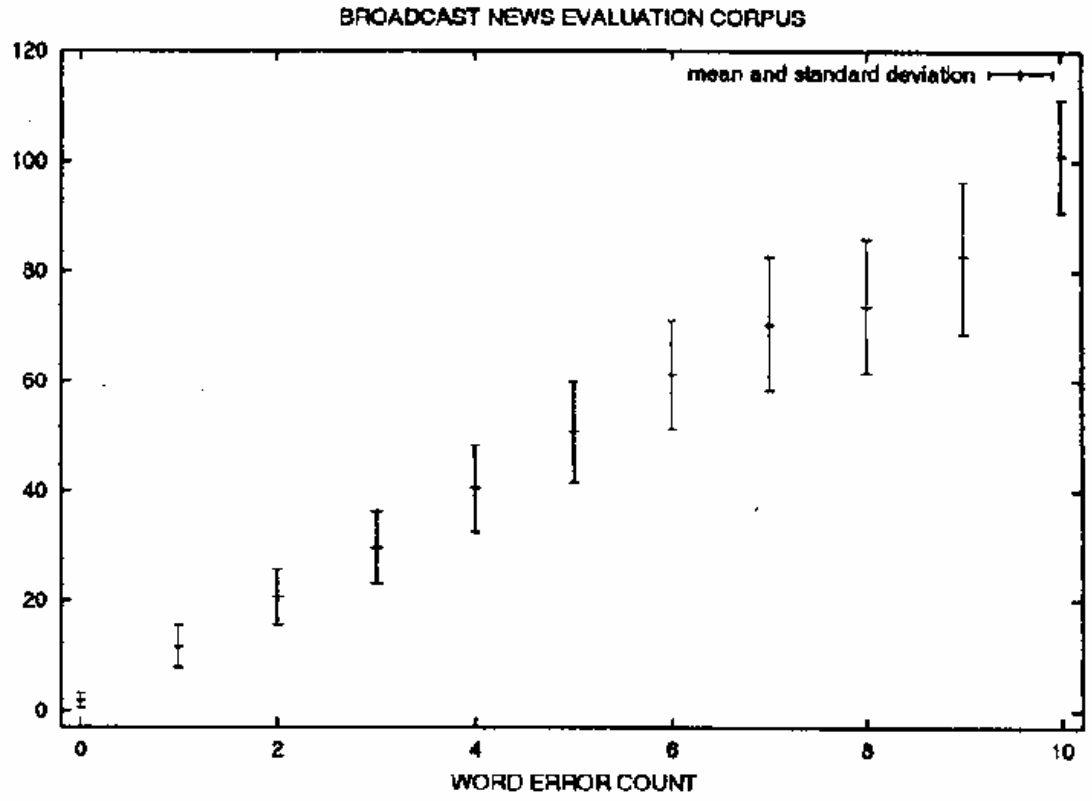


  - We can now define a now cost function using the notion of time frame errors

$$C(w_1^N, v_1^M) = \sum_{n=1}^{N} \frac{\sum_{t=s_n}^{t=e_n} 1 - \delta(w_n, v_t)}{1 + \alpha(e_n - s_n - 1)} \qquad (18)$$

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]

$$\{[w,s,e]_1^N\}_{opt} = \arg\min_{[w,s,e]_1^N}\left\{\sum_{[\nu,s',e']_1^M}\ell([w,s,e]_1^N,[\nu,s',e']_1^M)P([\nu,s',e']_1^M \mid x_1^T)\right\}$$

$$= \arg\min_{[w,s,e]_1^N}\left\{\sum_{[\nu,s',e']_1^M}\sum_{n=1}^{N}\frac{\sum_{t=s_n}^{t=e_n}1-\delta(w_n,\nu_t)}{1+\alpha(e_n-s_n-1)}P([\nu,s',e']_1^M \mid x_1^T)\right\}$$

$$= \arg\min_{[w,s,e]_1^N}\left\{\sum_{n=1}^{N}\frac{\sum_{t=s_n}^{t=e_n}\sum_{[\nu,s',e']_1^M}1\times P([\nu,s',e']_1^M \mid x_1^T)-\sum_{t=s_n}^{t=e_n}\sum_{[\nu,s',e']_1^M}\delta(w_n,\nu_t)P([\nu,s',e']_1^M \mid x_1^T)}{1+\alpha(e_n-s_n-1)}\right\}$$

$$= \arg\min_{[w,s,e]_1^N}\left\{\sum_{n=1}^{N}\frac{\sum_{t=s_n}^{t=e_n}\left[1-\sum_{[\nu,s',e']_1^M}\delta(w_n,\nu_t)P([\nu,s',e']_1^M \mid x_1^T)\right]}{1+\alpha(e_n-s_n-1)}\right\}$$

Some recent applications (9/9)

- Word error minimization (cont) :

$$\sum_{[\nu,s',e']_1^M} \delta(w_n,\nu_t) P([\nu,s',e']_1^M \mid x_1^T)$$

$$= \sum_{[\nu,s',e']_1^M} \sum_{\nu_m : s_m' \le t \le e_m'} \delta(w_n,\nu_t) P([\nu,s',e']_1^M \mid x_1^T)$$

$$= \sum_{[\nu,s',e'],\nu\ :s' \le t \le e'} \delta(w_n,\nu) P([\nu,s',e'] \mid x_1^T)$$

$$= p(w_n \mid t, x_1^T)$$

$$\frac{\sum_{t=s_n}^{t=e_n} \left[1 - p(w_n \mid t, x_1^T)\right]}{1 + \alpha(e_n - s_n - 1)}$$

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.