BIRS, Banff, Canada, 06-11 July 2009

Statistical Methods for Speech, Language and Image Processing: Achievements and Open Problems

Hermann Ney

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6
Computer Science Department
RWTH Aachen University, Germany

Outline

1 Overview

tasks considered:

- **speech recognition**
- **translation of text and speech**
- **image recognition: handwriting**

natural language processing: NLP
human language technology: HLT = NLP + speech

more tasks in HLT:

- **(spoken/written) language understanding**
- **dialog systems**
- **speech synthesis**
- **text summarization**
- **...**

Projects: TC-Star and GALE

TC-Star (2004-2007): integrated research project funded by EU:

- primary domain: Spanish/English speeches of EU parliament
  (TV station: Europe by satellite, 11 languages before EU extension)
- tasks: speech recognition, translation, synthesis
- partners:
  IBM, IRST Trento, LIMSI Paris, UKA Karlsruhe, UPC Barcelona, Siemens, ...
- first time: speech translation for real-life data

GALE (2005-2010?): funded by DARPA:

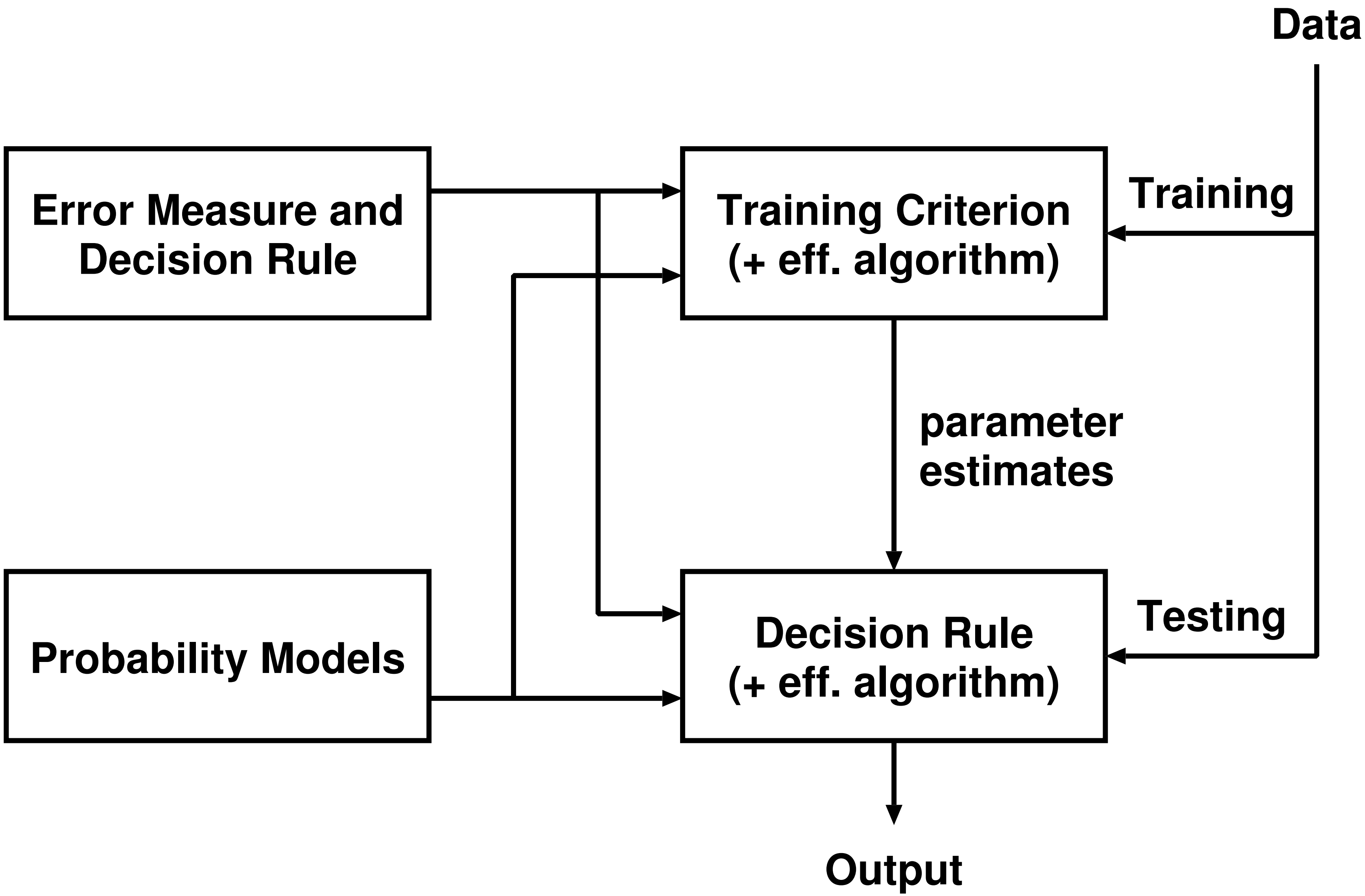
- primary domain:
  Arabic and Chinese, texts and TV shows
- three (huge) teams
  headed by BBN, IBM, SRI
- tasks: speech recognition, translation, information extraction

Joint Project QUAERO

QUAERO (2008-2013):
funded by OSEO/French Government

- **French partners and 2 German partners:**
  industry: Thomson, France Telecom, Jouve, LTU/Exalead, ...
  academia: CNRS, INRIA, INRA, U of Karlsruhe, RWTH Aachen, ...
- **primary goal: processing of**
  multimedia and multilingual documents (web, archives, ...)
- **many languages:**
  French, German, Chinese, Arabic,...
- **tasks:**
  speech recognition, translation of text and speech,
  handwriting recogition, image recognition, information extraction/retrieval, ...

four key ingredients:



2 Speech Recognition

- What are the main achievements over the last 30 years?
- What are the successful approaches?
- What are the lessons learned?

lessons:
– contribution from phonetics or linguistics: small
– data-driven methods
– avoid local decisions
– consistent models and training criteria
– comparative evaluations

public software for ASR: RWTH i6 web site

2.1 Problem of Speech Recognition

characteristic properties:

- **high variability:** – from utterance to utterance
  – dependence on the phonetic context
  – from speaker to speaker
- **speaking rate:** – can vary drastically
  – no anchor points
- **word and phoneme (sound) boundaries:** do not exist in acoustic signal
- **context or prior information:**
  syntactic-semantic structures of the spoken language

compare:

- **human-human communication:**
  – proper names via telephone $\rightarrow$ spelling
  – native language $\rightarrow$ foreign language:
  $\rightarrow$ understanding much harder
- **character recognition:**
  character error rate: 20–30% [sub.+del.+ins.]

State of the Art

heavy dependence on:
– vocabulary size
– perplexity of LM
– speaking style
– acoustic quality

standard data bases
with following conditions:
– American English
– continuous speech
– speaker independent
– 100 hours of speech and more

performance of best research systems:

Task	Speaking Style	Vocabulary Size	Perplexity	Word Error Rate [%]
Digit Strings	read	11	11	0.3
Voice Commands	read	1000	60	6.0
Text dictation	read	64 000	150	10.0
Broadcast News	natural	64 000	200	15.0
Telephone Conversations	colloquial	64 000	120	30.0

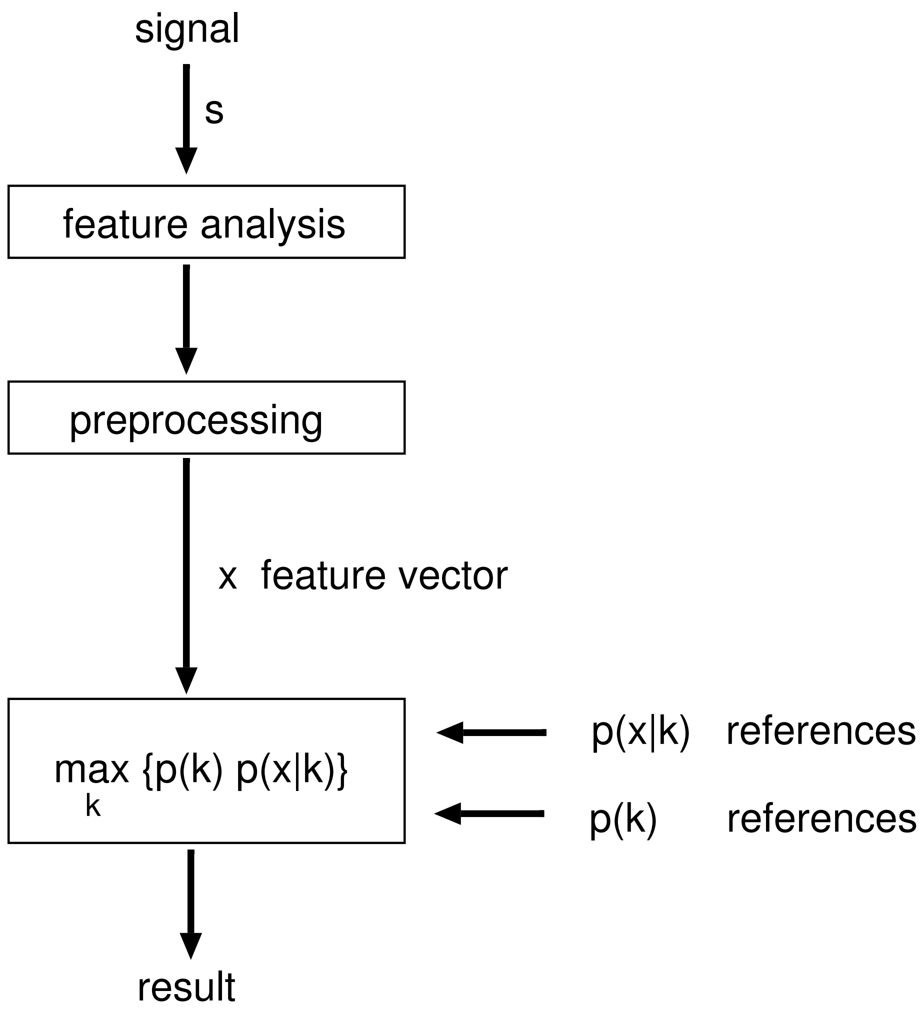
2.2 Bayes Decision Rule

goal:
minimize the decision errors
$\rightarrow$ Bayes decision rule:

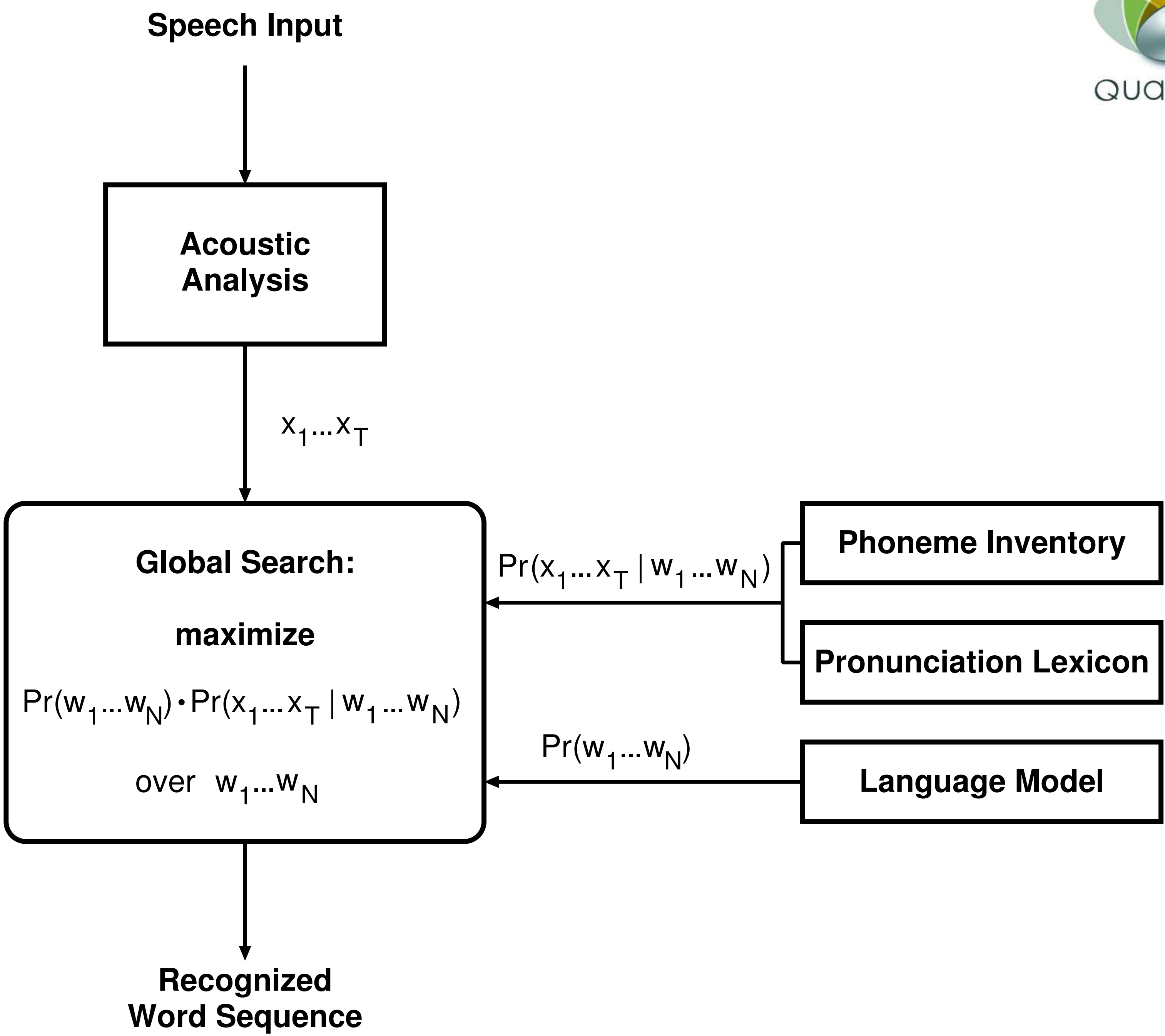
$$x \rightarrow k(x) := \operatorname*{argmax}_k \{p(k|x)\}$$
$$= \operatorname*{argmax}_k \{p(k) \cdot p(x|k)\}$$

holistic approach
to speech recognition:

– **class k:**
  word sequence $w_1^N = w_1 ... w_N$
– **observation x:**
  vector sequence $x_1^T = x_1 ... x_T$



Speech Recognition: Bayes Decision Rule



2.3 Acoustic Modelling

Problem:

prob.distributions over sequences w_1^N and x_1^T
$\longrightarrow$ factorization of probability distributions

Hidden Markov Model (HMM):

$$\begin{aligned} Pr(x_1^T|w_1^N) &= \sum_{s_1^T} Pr(x_1^T, s_1^T|w_1^N) = \sum_{s_1^T} \prod_{t=1}^{T} p(x_t, s_t|s_{t-1}, w_1^N) \\ &= \sum_{s_1^T} \prod_{t=1}^{T} \left[p(s_t|s_{t-1}, w_1^N) \cdot p(x_t|s_t, w_1^N) \right] \end{aligned}$$

HMM at several levels:
– phoneme
– word: concatenation of phonemes
– sentence: concatenation of words

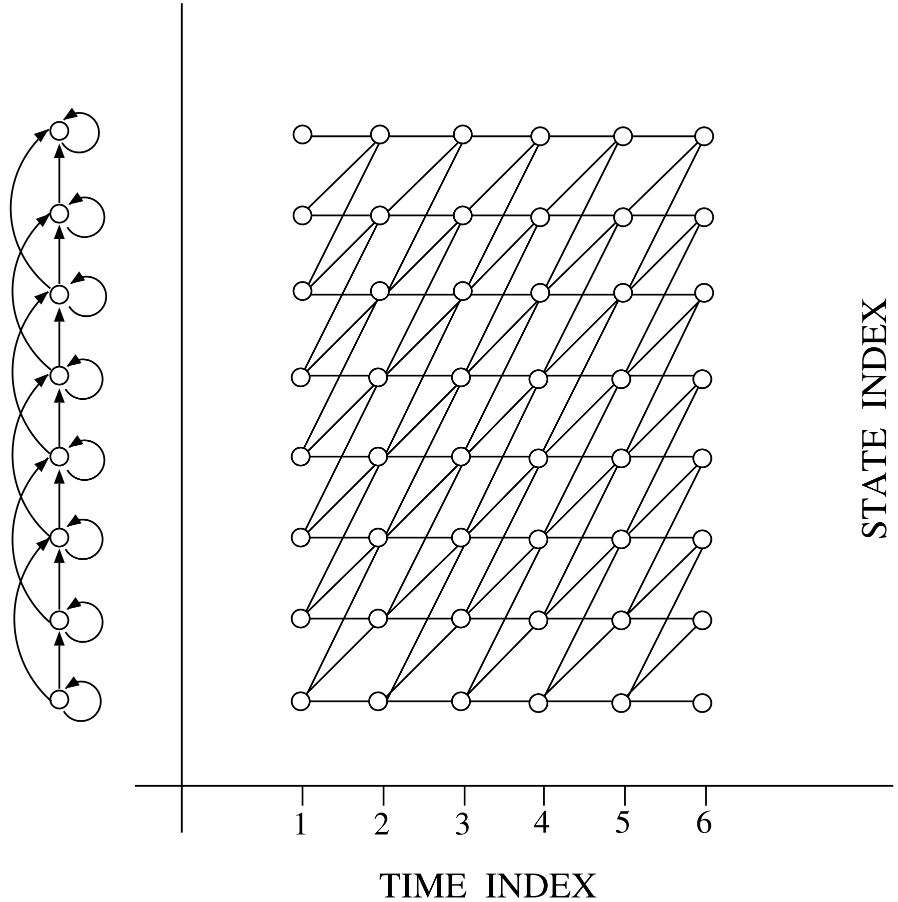
Hidden Markov Model (HMM)

HMM = statistical finite-state automaton (phoneme, word, sentence) with first-order dependencies

- **observations: continuous-valued vectors x_t**
- **two distributions:**
  $p(x_t, s|s', w) = p(s|s', w) \cdot p(x_t|s, w)$:
  - **transition prob. $p(s|s', w)$: prior model structure**
  - **emission prob. $p(x_t|s, w)$: link to observations**

note:

- **efficient probability model $p(x_1...x_T|w)$ for string $x_1...x_T$**
- **handling of time alignment problem**



Refinements

- **augmented vector: window around t:**

$$p(x_t|s,w) \text{ with } x_t = [z_{t-\delta}, ..., z_t, ..., z_{t+\delta}]$$

  with original acoustic vectors x_t over time t

- **LDA: linear discriminant analysis for reducting the dimension of the augmented feature space**

- **Gaussian mixture (multimodal distribution):**

$$p(x|s,w) \; = \; \sum_i p(x,i|s,w) = \sum_i p(i|s,w) \; p(x|s,i,w)$$

- **phoneme models in triphone context: decision trees (CART) for finding equivalence classes**

2.4 Baseline Training

- **key quantity: class posterior probability**

$$p_\theta(c|x) \;=\; \frac{p_\theta(c)\; p_\theta(x|c)}{\sum_{c'} p_\theta(c')\; p_\theta(x|c')}$$

with parameter set θ to be trained

- **natural criterion with labeled training data $(x_r, c_r), r = 1, ..., R$:**

$$\arg\max_\theta \left\{ \sum_r \log\;\; p_\theta(c_r|x_r) \right\}$$

- **approximation: numerator only = joint likelihood 'maximum-likelihood' (in engineering, pattern recognition, ...)**

$$\arg\max_\theta \left\{ \sum_r \log\;\; p_\theta(c_r, x_r) \right\} =$$

$$= \;\; \arg\max_\theta \left\{ \sum_r \log\;\; p_\theta(c_r) + \sum_r \log\;\; p_\theta(x_r|c_r) \right\}$$

- **additional complication:**
  - **HMM with hidden variables**
  - **EM algorithm and its variants**

2.5 Training using the EM Algorithm

define model distribution $p_\vartheta(y|x)$
for random variables x and y and parameters ϑ
and hidden variable A:

$$\begin{aligned} p_\vartheta(y|x) &= \sum_A p_\vartheta(y, A|x) \\ &= \sum_A p_\vartheta(A|x) \cdot p_\vartheta(y|A, x) \end{aligned}$$

examples of hidden variables:
– mixture index in Gaussian mixtures
– linear interpolation for language models
– time alignment in HMM for ASR
– word alignment in HMM and IBM-2 for SMT

training data:

$$(x_n, y_n), \quad n = 1, ..., N$$

training criterion: consider the likelihood function (or its logarithm)
and maximize it over the unknown parameters ϑ:

$$\begin{aligned}\vartheta \;\rightarrow\; F(\vartheta) \;&:=\; \log \prod_n \sum_A p_\vartheta(y_n, A|x_n) \\ &=\; \sum_n \log \sum_A p_\vartheta(y_n, A|x_n)\end{aligned}$$

typical situation:
– no closed-form solution
– iterative procedures using the EM algorithm

Derivation of EM Algorithm

For the difference of log-likelihoods with two parameter estimates ϑ and $\hat{\vartheta}$, we have the following inequality:

$$F(\hat{\vartheta}) - F(\vartheta) \;\geq\; Q(\vartheta;\hat{\vartheta}) - Q(\vartheta;\vartheta)$$

with the definition of the $Q(\cdot;\cdot)$ function:

$$Q(\vartheta;\hat{\vartheta}) \;:=\; \sum_n \sum_A \gamma_n(A|\vartheta)\;\log\, p_{\hat{\vartheta}}(y_n, A|x_n)$$

and the (sort of) posterior probabilities $\gamma_n(A|\vartheta)$:

$$\gamma_n(A|\vartheta) \;:=\; p_\vartheta(A|x_n, y_n)$$

$$= \frac{p_\vartheta(y_n, A|x_n)}{\sum_{A'} p_\vartheta(y_n, A'|x_n)}$$

proof of inequality: based on divergence inequality (see literature)

EM Algorithm

operations of EM algorithm:

$$\hat{\vartheta} \;:=\; \operatorname*{argmax}_{\vartheta} \left\{ \sum_n \sum_A \gamma_n(A|\vartheta) \;\log\, p_{\hat{\vartheta}}(y_n, A|x_n) \right\}$$

- **E = expectation of** $\log\, p_{\hat{\vartheta}}(y_n, A|x_n)$
- **M = maximization of** $Q(\vartheta; \hat{\vartheta})$ **over** $\hat{\vartheta}$
  (most attractive if there is a closed-form solution!)

EM algorithm = iterative procedure:

- **previous estimate:** ϑ
- **new estimate:** $\hat{\vartheta}$
  (local convergence is guaranteed)

EM algorithm: interpretation:

- **weighted likelihood function for the model** $p(y, A|x)$
- **weights = posterior probabilities** $\gamma_n(A|\vartheta)$

Maximum Approximation

exact criterion:

$$\hat{\vartheta} \;=\; \arg\max_{\vartheta} \left\{ \sum_{n} \log \sum_{A} p_{\vartheta}(y_n, A | x_n) \right\}$$

maximum approximation (Viterbi training): replace sum by maximum:

$$\hat{\vartheta} \;\cong\; \arg\max_{\vartheta} \left\{ \sum_{n} \log \max_{A} p_{\vartheta}(y_n, A | x_n) \right\}$$

iterative procedure with the alternating steps:

$$\hat{\vartheta} \;:=\; ...$$

$$\hat{A}_n \;:=\; \arg\max_{A} \left\{ p_{\hat{\vartheta}}(y_n, A | x_n) \right\} \qquad n = 1, ..., N$$

$$\hat{\vartheta} \;:=\; \arg\max_{\vartheta} \left\{ \sum_{n} \log \, p_{\hat{\vartheta}}(y_n, \hat{A}_n | x_n) \right\}$$

$$\hat{A}_n \;:=\; ...$$

2.6 Language Model

Trigram Model (for sentence prior):

$$Pr(w_1^N) = \prod_{n=1}^{N} p(w_n|w_1^{n-1}) = \prod_{n=1}^{N} p(w_n|w_{n-2}, w_{n-1})$$

Disambiguation of Homophones:

- **Homophones: two, to, too**

  Twenty-two people are too many to be put in this room.

- **Homophones: right, write, Wright**

  Please write to Mrs. Wright right away.

- **problem: unseen events:**
  $64\,000$ **words:** $64\,000^3 = 2^{18} \cdot 10^9$ **trigrams**

  consequence: virtuall all word trigrams
  have relative frequeny = 0

- **remedy: smoothing**

- **leave-one-out (or cross-validation)**
  – empirical Bayes estimate
  – Turing-Good estimate

2.7 Search

Search or Decoding:

$$\arg\max_{w_1^N, s_1^T} \left\{ \prod_n p\Big(w_n | w_{n-2}, w_{n-1}\Big) \cdot \prod_t \; p(s_t | s_{t-1}, w_1^N) \cdot p(x_t | s_t, w_1^N) \right\}$$

- **consequence: holistic approach**
  - **no segmentation**
  - **no local decisions**
  - **time alignment is part of decision process**
- **search strategy: dynamic programming with refinements**
  - **beam search and pruning**
  - **look-ahead estimates**
  - **word lattice rather than single best sentence**

2.8 Adaptation

adaptive recognition:

- **recognition problem may depend on varying conditions: room acoustics, speaker, microphone, ...**
- **Bayesian spirit: assume adaptation parameter set α and integrate out α (with $X = x_1^T, W = w_1^N$):**

$$\begin{aligned} p(X|W,\theta) &= \int d\alpha\; p(X,\alpha|W;\theta) \\ &= \int d\alpha\; p(\alpha|W,\theta) \cdot p(X|W;\theta,\alpha) \\ &\cong \max_{\alpha} \Big\{ p(\alpha|W,\theta) \cdot p(X|W;\theta,\alpha) \Big\} \end{aligned}$$

- **Bayes decision rule:**

$$\arg\max_{W} \Big\{ p(W) \cdot \max_{\alpha} p(X,\alpha|W;\theta) \Big\}$$

Adaptation: Impact on Architecture

- **recognition:**

$$\arg\max_{W} \left\{ p(W) \cdot \max_{\alpha} p(X, \alpha | W; \theta) \right\}$$

  implementation: estimate α in
  a) two recognition passes
  b) text-independent mode

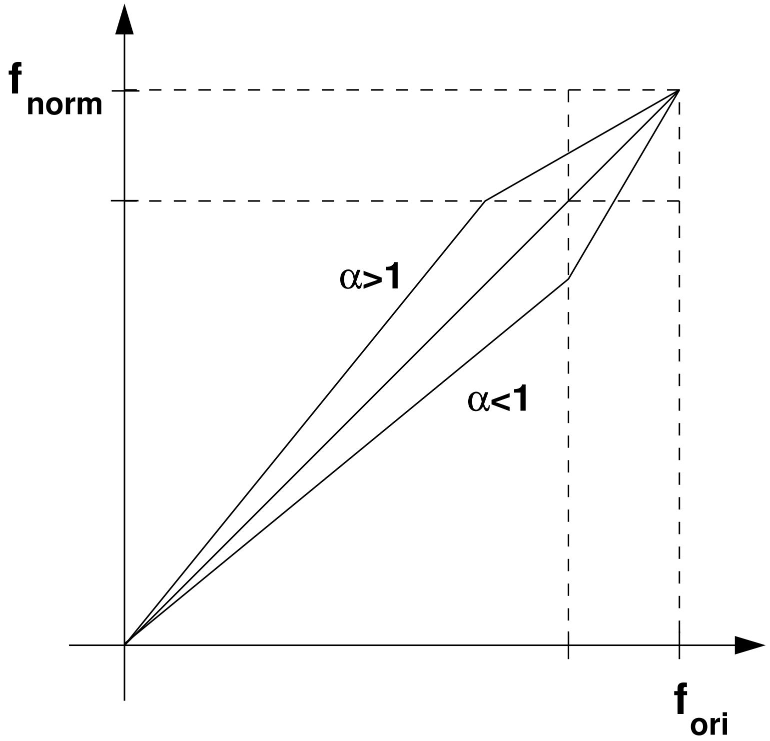
- **training**
  with training data (X_r, W_r) for each speaker $r = 1, ..., R$:

$$\arg\max_{\theta} \prod_{r=1}^{R} \max_{\alpha} \left\{ p(X_r, \alpha | W_r; \theta) \right\}$$

  result: more complex optimization problem

Adaptation: VTN (= Vocal Tract Normalization)

- vocal tract length:
  – depends on speaker
  – irrelevant for recognition
- approach:
  – 'linear' scaling (warping) of frequency axis
  – reference model: normalized frequency axis



Remarks:

- VTN: frequency axis normalization
- Wakita 1975-77

other types of adaptation:
linear transformation: matrix

2.9 Discriminative Training

notation:

r: sentence index
X_r: sequence of feature vectors of sentence r
W_r: spoken word sequence of sentence r,
W: any word sequence

- **class posterior probability:**

$$F(\theta) = \sum_r \log p_\theta(W_r|X_r) \qquad p_\theta(W_r|X_r) := \frac{p(W_r)p_\theta(X_r|W)}{\sum_W p(W)p_\theta(X_r|W)}$$

- **MCE: minimum classification error rate ('old' concept in pattern recognition):**

$$F(\theta) = \sum_r \frac{1}{1 + \left(\frac{p(W_r)p_\theta(X_r|W_r)}{\max\limits_{W \neq W_r} p(W)p_\theta(X_r|W)}\right)^{2\beta}}$$

(β: smoothing constant)

discriminative training: practical aspects

- **initialization:**
  acoustic models trained by Max.Lik.

- **implementation details:**
  – word lattice (for sum in denominator)
  – unigram LM in training
  – scaling of acoustic and language models

- **experimental results:**
  – MCE: typically better
  – discriminative training more efficient after adaptation (SAT)

2.10 Results

effects of specific methods in RWTH system (WER [%]):

	English		Spanish	
	dev06	eval06	dev06	eval06
baseline	15.7	13.1	9.9	13.8
+ adaptation (SAT)	14.0	11.5	7.9	-
+ unsupervised data	12.9	-	-	-
+ discrim. training	12.5	-	7.3	9.6
+ adaptation (MLLR)	11.8	9.8	7.1	9.3
+ improved lexicon	11.6	9.6	7.1	9.3
+ larger LM	11.0	8.5	-	-
+ system combination	10.6	8.4	-	-

rule of thumb: reduction of WER
by one third over baseline system

word error rates [%]:

System	open	public	restricted
RWTH			8.9
LIMSI			9.2
IBM	9.2		9.4
IRST		9.6	9.5
LIUM			19.8
UPC			27.5
DAEDALUS		46.6	
SysComb		7.4	

TC-Star Evaluation 2007: English

word error rates [%]:

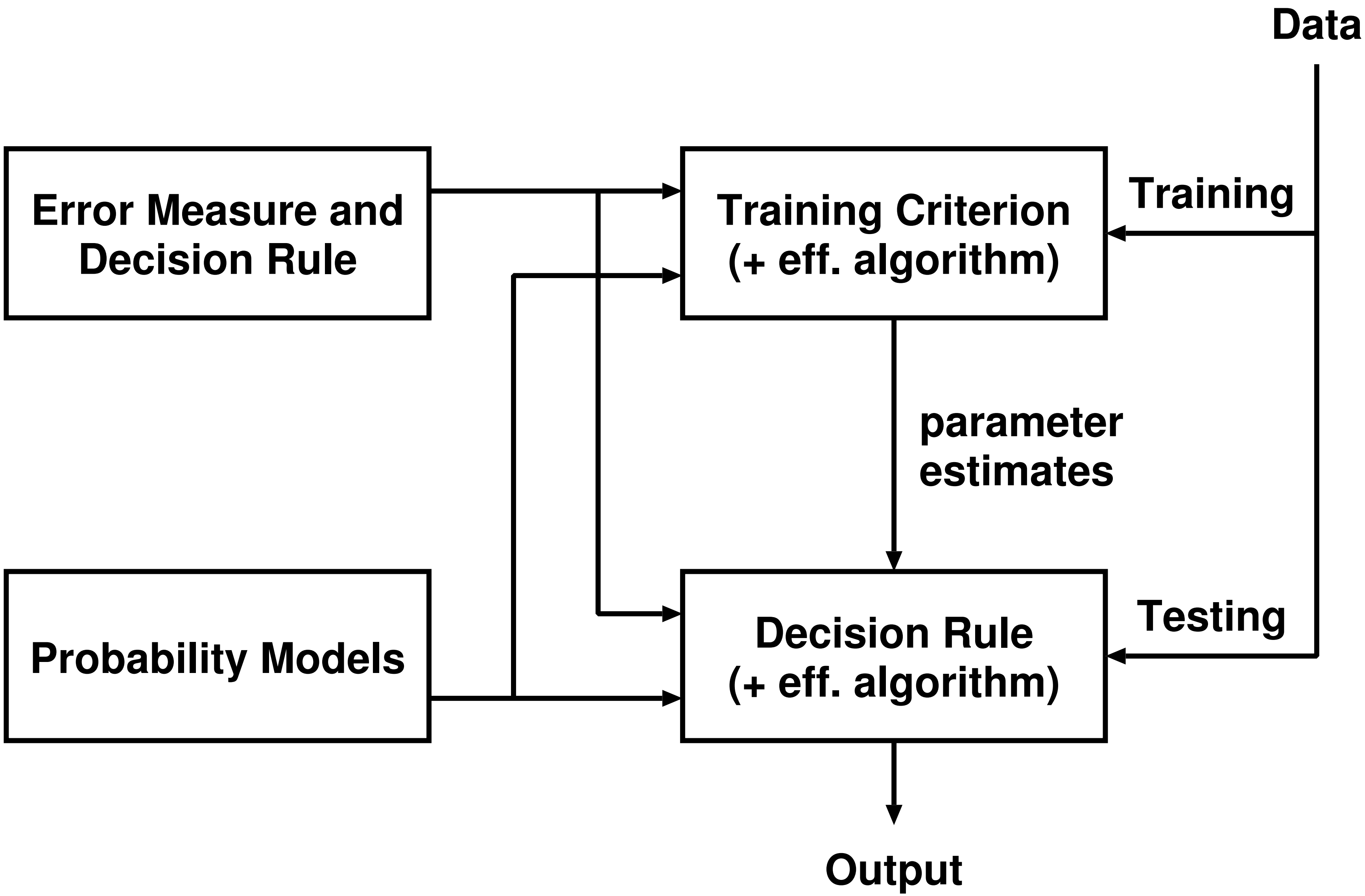
	open	public	restricted
RWTH		9.0	9.7
LIMSI		9.1	
IBM	7.1	9.2	9.8
UKA		9.2	
IRST		10.2	11.3
LIUM		22.1	22.4
SysComb		6.9	

2.11 The Statistical Approach Revisited

two attractive properties:

- **holistic decision criterion:**
  - **exploits all (available) dependencies (= knowledge sources)**
  - **is able to combine thousands/millions of weak dependencies**
  - **handles interdependences, ambiguities and conflicts**
- **powerful training methods:**
  - **training criterion is (ideally!) linked to PERFORMANCE**
  - **fully AUTOMATIC procedures (no human involved !)**
  - **HUGE amounts of data can be exploited**

four key ingredients:



Four Key Components

- **error measure (= loss function) and decision rule:**

$$R(c|x) \;:=\; \sum_{\tilde{c}} pr(\tilde{c}|x)\, L[c,\tilde{c}]$$

$$x \rightarrow \hat{c}(x) \;=\; \arg\min_{c} \Big\{ R(c|x) \Big\}$$

  with a suitable definition of $L[c,\tilde{c}]$

- **probability model $p_\theta(c|x)$ or $p_\theta(c) \cdot p_\theta(x|c)$**
  is used to replace $pr(c|x)$ or $pr(c) \cdot pr(x|c)$

- **training criterion (+ eff. algorithm)**
  to learn the unknown parameters θ from training data

- **decision rule (+ eff. algorithm) :**
  search or decoding: requires optimization (sometimes hard!)

Error Measures and Decision Rules: Exmples

inconsistencies:

- **POS tagging:**
  - **in practice: symbol error rate**
  - **in Bayes rule: 0/1 loss (= SER, sentence error)**
- **speech recognition:**
  - **in practice: edit distance (= WER, word error rate)**
  - **in Bayes rule: 0/1 loss (= SER, sentence error)**
- **machine translation:**
  - **in practice: BLEU or TER (= translation error rate)**
  - **in Bayes rule: 0/1 loss (= SER, sentence error)**

attempts to go beyond 0/1 loss function:
only small or negligible improvements

Probability Models: Exmples

- **discriminant functions (linear and nonlinear)**
- **neural networks: (virtually) any structure**
- **Gaussian classifier**
- **Gaussian mixtures**
- **models with hidden variables (path, alignment):**
  - **Hidden Markov models (HMM) in speech recognition**
  - **alignment models in machine translation**
- **maximum entropy models (log-linear, exponential, multiplicative)**
- **decision trees (CART)**
- **...**

Training Criteria (+ Eff. Algorithms): Examples

labelled training data: $(x_r, c_r), r = 1, ..., R$

- **maximum likelihood:**

$$\arg\max_{\theta} \left\{ \sum_{r=1}^{R} \log\, p_\theta(c_r) \right\} \qquad \textbf{and} \qquad \arg\max_{\theta} \left\{ \sum_{r=1}^{R} \log\, p_\theta(x_r|c_r) \right\}$$

- **posterior probability (or MMI):**

$$\arg\max_{\theta} \left\{ \sum_{r=1}^{R} \log\, p_\theta(c_r|x_r) \right\}$$

- **squared error criterion:**

$$\arg\min_{\theta} \left\{ \sum_{r=1}^{R} \sum_{c} \Big[p_\theta(c|x_r) - \delta(c, c_r) \Big]^2 \right\}$$

- **minimum classification error (MCE, smoothed error count)**
- **...**

- **EM (expectation/maximization) algorithm:**
  maximum likelihood for hidden-variable models
  (maximum approximation: Viterbi training)

- **error back propagation:**
  squared error criterion for neural networks

- **GIS (general iterative scaling):**
  posterior probability for maximum entropy (log-linear) models

- **...**

Decision Rule (+ Eff. Algorithms): Examples

specific algorithms depends on probability models:

- **forward algorithm:**
  for HMM in speech recognition (for a single hypothesized word sequence)

- **dynamic programming**
  – for POS tagging and other tagging tasks:
  – for small vocabulary-speech recognition,
  – for translation using finite-state transducers

- **time-synchronous beam search and A* search:**
  for large-vocabulary speech recognition

- **position-synchronous beam search and A* search:**
  for large-vocabulary language translation

- **...**

Why does a statistical decision system make errors?

To be more exact:
Why errors IN ADDITION to the minimum Bayes errors?

Reasons from the viewpoint of Bayes' decision rule:

- **incorrect input (or observation):**
  only an incomplete part or a poor transformation of the true observations is used.
- **incorrect modelling:**
  - **incorrect probability distribution**
  - **not enough training data**
  - **poor training criterion**
  - **convergence problems: slow or several optima**
- **incorrect search or generation:**
  - **suboptimal decision rule**
  - **suboptimal search procedure**

open issues:

- **symbol vs. string error rate: inconsistencies in Bayes decision rule: theoretical justification for negative experimental results?**
- **need for better features and dependencies in acoustic models:**
  $\rightarrow$ improved robustness of ASR systems
- **various aspects in discriminative training:**
  – various criteria: MMI vs. sentence, word, phone error rate
  – problems with local optima
  – efficient optimization strategies
- **within discriminative training:**
  signal analysis and feature extraction

3 Discriminative Models, Log-linear Models and CRFs

3.1 Motivtion

- log-linear models: well known in statistics
  advantage: convex optimization problem in training
- recent results by Heigold et al. (RWTH Aachen)
  (Eurospeech'07, ICASSP'08, ICML'08, Interspeech'08):
  class posterior of many generative models = log-linear model or CRF
- experimental results:
  ongoing work

HMM in ASR:

- **observations:**
  - **sequence of acoustic vectors** x_1^T
  - **sequence of words** w_1^N
- **hidden variables: state sequence (= alignment path)** s_1^T

$$\begin{aligned} p(w_1^N, x_1^T) &= p(w_1^N)\, p(w_1^N | x_1^T) \\ &= p(w_1^N) \sum_{s_1^T} p(s_1^T | w_1^N)\, p(x_1^T | w_1^N, s_1^T) \\ &\quad \textbf{(assumption: first-order dependencies)} \\ &= p(w_1^N) \sum_{s_1^T} \prod_t p(s_t | s_{t-1}, w_1^N)\, p(x_t | s_t, w_1^N) \end{aligned}$$

discriminative training: posterior probability of word sequence:

$$\begin{aligned} p(w_1^N|x_1^T) &= \sum_{s_1^T} p(w_1^N, s_1^T|x_1^T) \\ &= \frac{1}{p(x_1^T)} \cdot p(w_1^N) \sum_{s_1^T} \prod_t p(s_t|s_{t-1}, w_1^N)\, p(x_t|s_t, w_1^N) \\ &= \frac{1}{p(x_1^T)} \cdot p(w_1^N) \sum_{s_1^T} \exp\Big(\sum_t [\log\, p(s_t|s_{t-1}, w_1^N) + \log\, p(x_t|s_t, w_1^N)]\Big) \end{aligned}$$

relation to log-linear model/CRF will be studied in 3 steps:

- **frame level: Gaussian model**
- **sequence level:**
  - without alignments
  - with alignments: maximum approximation

3.2 Frame Level

frame level: Gaussian model for $x \in \mathrm{I\!R}^{\mathrm{D}}$ and class c:

$$\begin{aligned}
p_\theta(c|x) &= \frac{1}{p(x)} \cdot p(c)\,\mathcal{N}(x|\mu_c, \Sigma_c) \\
&= \frac{1}{p(x)} \cdot \frac{p(c)}{\sqrt{\det(2\pi\Sigma_c)}} \exp\Big(-\frac{1}{2}(x-\mu_c)^t \Sigma_c^{-1} (x-\mu_c) \Big) \\
&= \frac{1}{p(x)} \cdot \exp\Big(\log p(c) - \frac{1}{2}\log\det(2\pi\Sigma_c) - \frac{1}{2}\mu_c^t \Sigma_c^{-1}\mu_c + \mu_c^t \Sigma_c^{-1} x - \frac{1}{2} x^t \Sigma_c^{-1} x \Big) \\
&= \frac{1}{p(x)} \cdot \exp\Big(\alpha_c + \lambda_c^T x + x^T \Lambda_c x \Big)
\end{aligned}$$

with the (constrained) parameters:

$$\theta := \{\alpha_c \in \mathrm{I\!R}, \quad \lambda_c \in \mathrm{I\!R}^{\mathrm{D}}, \quad \Lambda_c \in \mathrm{I\!R}^{\mathrm{D \cdot D}}\}$$

important result: log-linear model:
– (log) linear in parameters $\alpha_c \in \mathrm{I\!R}, \lambda_c \in \mathrm{I\!R}^{\mathrm{D}}, \Lambda_c \in \mathrm{I\!R}^{\mathrm{DxD}}$
– (log) quadratic in observations x

posterior form of Gaussian model:

- **log-linear model is invariant under additive transformations:**

$$
\begin{aligned}
\alpha_c &\rightarrow \alpha_c + \alpha_0 &\in \mathbb{R} \\
\lambda_c &\rightarrow \lambda_c + \lambda_0 &\in \mathbb{R}^{\mathrm{D}} \\
\Lambda_c &\rightarrow \Lambda_c + \Lambda_0 &\in \mathbb{R}^{\mathrm{DxD}}
\end{aligned}
$$

- **for conversion back to Gaussian model:**
  exploit these invariances to satisfy the constraints of Gaussian model:
  – normalization of $p(c)$
  – positive definite property of Σ_c
  – invertibility of Σ_c

- **note when going generative Gaussian model to its posterior form:**
  parameters of Gaussian model are not unique anymore!

Equivalence

result: EXACT equivalence between

- **posterior form of Gaussian model**
- **log-linear model with quadratic observations (features)**

consequence:
discriminative training criterion for Gaussian models defines a convex optimization problem:

$$\arg\max_{\theta} \left\{ \sum_{r} \log\ p_{\theta}(c_r|x_r) \right\}$$

with labelled training data $(x_r, c_r), r = 1, ..., R$

High-Order Features

generalization: define high-order features $y \in \mathbb{R}^{D_y}$ for $x \in \mathbb{R}^D$:

$$
\begin{aligned}
y &:= [1, x_1, ..., x_d, ..., x_D, x_1^2, ..., x_{d_1} x_{d_2}, ..., x_D^2, x_1^3, ..., x_{d_1} x_{d_2} x_{d_3}, ..., x_D^3, ...] \\
D_y &:= D^0 + D^1 + D(D+1)/2! + D(D+1)(D+2)/3! + ...
\end{aligned}
$$

or more general feature function:

$$
x = x_1^D \rightarrow y = y(x) \in \mathbb{R}^{D_y}
$$

log-linear model for class posterior probability:

$$
\begin{aligned}
p(c|x) &= p(c|y) \\
&= \frac{\exp\left[\lambda_c^t y\right]}{\sum\limits_{c'} \exp\left[\lambda_{c'}^t y\right]}
\end{aligned}
$$

properties of training criterion:

- **convex problem**
  (proof: compute and consider second derivative)
- **no closed form solution**
- **strategy: solve the optimization problem directly**
  (not the equaton using the derivatives)
- **convergence might be very slow**
- **parameters may not be unique,**
  but the posterior model is!
- **overfitting: (some) parameters might tend to $\pm\infty$**
  $\rightarrow$ remedy: regularization

3.3 String Level without Alignments

example of string handling: POS tagging problem

- **observations:**
  sequence of (written) words: $x_1^N = x_1, ..., x_n, ..., x_N$
- **goal: for each word position** n**,**
  find the associated POS label c_n **to form the tag sequence** $c_1^N = c_1, ..., c_n, ..., c_N$

compare with notation in speech HMM:
  word sequence: x_1^T := **sequence of acoustic observations**
  label sequence: s_1^T := **sequence of states**

generative model: POS bigram model ('HMM approach'):

- **generative model with the joint probability:**

$$p(c_1^N, x_1^N) \;=\; \prod_n \left[p(c_n|c_{n-1})\, p(x_n|c_n)\right]$$

  with membership probability $p(x|c) = p(x_n|c_n)$ **and bigram model** $p(c|c') = p(c_n|c_{n-1})$

- **free parameters of model:**
  entries of tables $p(x|c)$ **and** $p(c|c')$

consider the posterior form of this model:

$$\begin{aligned} p(c_1^N|x_1^N) &= \frac{p(c_1^N, x_1^N)}{\sum_{\tilde{c}_1^N} p(\tilde{c}_1^N, x_1^N)} = \frac{1}{p(x_1^N)} \cdot p(c_1^N, x_1^N) \\ &= \frac{1}{p(x_1^N)} \cdot \prod_n p(c_n|c_{n-1})\, p(x_n|c_n) \end{aligned}$$

convert to log-linear form:

$$\begin{aligned} p(c_1^N|x_1^N) &= \frac{1}{p(x_1^N)} \cdot \exp\Big(\sum_n \big[\log p(c_n|c_{n-1}) + \log p(x_n|c_n) \big] \Big) \\ &= \frac{1}{p(x_1^N)} \cdot \exp\Big(\sum_n \big[\lambda(c_n; c_{n-1}) + \lambda(x_n; c_n) \big] \Big) \end{aligned}$$

which is the form of a CRF (conditional random field)

constraints: normalization requirements
– experiments: do they matter?
– theory: do they cancel?

Mathematical Equivalence

consider modified model:

- **add string end symbol \$ to tag set Σ**
- **normalization constraint:**

$$\sum_{c \in \Sigma \cup \{\$\}} p(c|c') = 1 \qquad \forall c' \in \Sigma \cup \{\$\}$$

- **experimental check:**
  no degradation in performance due to modification

mathematical analysis using matrix algebra (Heigold Interspeech'08):

posterior form of POS bigram tagging model
is a log-linear model (or CRF)

more precise terminology for CRF in speech and language processing:
one-dimensioal CRF with log-linear first-order dependencies

3.4 HMM: State Level with Alignments

posterior probability of word sequence:

$$p(w_1^N|x_1^T) \;=\; \frac{1}{p(x_1^T)} \cdot \sum_{s_1^T} \exp\left(\log p(w_1^N) + \sum_t [\log\, p(s_t|s_{t-1}, w_1^N) + \,\log\, p(x_t|s_t, w_1^N)]\right)$$

assumption: estimate state sequence by maximum approximation

joint posterior probability of word and state sequence:

$$p(w_1^N, s_1^T|x_1^T) \;=\; \frac{1}{p(x_1^T)} \cdot \exp\left(\log\, p(w_1^N) \,+\, \sum_t [\log\, p(s_t|s_{t-1}, w_1^N) + \,\log\, p(x_t|s_t, w_1^N)]\right)$$

which, for Gaussian emission models, will be an exact log-linear model!

summary: discriminative training of HMMs:

- **conventional training criterion ('MMI')**
- **with known state sequence: convex problem**
- **maximum approximation:**
  – alternating optimization between alignment and parameter learning
  – only LOCAL convergence
- **attractive property:**
  ALL parameters of the model can be trained:
  Gaussian parameter, transition probabilities, LM scale factor, ...

key problem: efficient calculation of the denominator
– even polynomial complexity might require numeric approximations
– approximations to sum: word lattice or beam search

3.5 C^4: Correctness, Complexity, Convexity, Convergence

most important aspect: correctness of training criterion:

e.g. criterion: $\log\ p(c|x)$ **vs.** $\log\ p(x|c)$

various level of training complexity:

- **closed-form solutions:**
  typical example: max.lik. estimation
  (for Gaussian, Poisson, multinomial models)
- **convex, without closed-form solution:**
  – **typical examples: SVM and log-linear models**
  – **advantage: no problem with initialization**
- **local optimum, with guaranteed convergence:**
  – **typical examples: EM for Gaussian mixtures and HMM, K-Means with splitting, Hidden-GIS algorithm [Heigold ICASSP 08]**
  – **advantage: no problems with step size**
- **local optimum with explicit gradient**
  – **convergence must be controlled via step-size**

4 Statistical MT

4.1 History

use of statistics has been controversial in NLP
(NLP := natural language processing):

- Chomsky 1969:
  ... the notion ’probability of a sentence’ is an entirely useless one,
  under any known interpretation of this term.
- was considered to be true by most experts in NLP and AI

1988: IBM starts building a statistical system for MT (= machine translation)
(in opposition to linguistics and artificial intelligence)

short (and simplified) history:

- **1949 Shannon/Weaver: statistical (=information theoretic) approach**
- **1950–1970 empirical/statistical approaches to NLP ('empiricism')**
- **1969 Chomsky: ban on statistics in NLP**
- **1970–? hype of AI and rule-based approaches;**
  BUT: statistical methods for speech recognition
- **1988–1995 statistical translation at IBM Research:**
  - **corpus: Canadian Hansards: English/French parliamentary debates**
  - **DARPA evaluation in 1994:**
    comparable to 'conventional' approaches (Systran)
- **1992 workshop TMI: *Empiricist vs. Rationalist Methods in MT***
  controversial panel discussion

Beyond IBM: 1992 – 2008

limited domain (data collected in lab):

- **speech translation:**
  travelling, appointment scheduling,...
- **projects:**
  – C-Star consortium
  – Verbmobil (German)
  – EU projects: Eutrans, PF-Star

'unlimited' domain (real-life data):

- **US DARPA TIDES 2001-04: written text (newswire):**
  Arabic/Chinese to English
- **EU TC-Star 2004-07: speech-to-speech translation**
- **US DARPA GALE 2005-2010:**
  – Arabic/Chinese to English
  – speech and text
  – ASR, MT and information extraction

automatic speech recognition (ASR): key ideas:

- **Bayes decision rule:**
  – minimizes the decision errors
  – defines the probabilistic framework

- **probabilistic structures**
  – problem-specific models (in lieu of ’big tables’)
  – strings: hidden variables (alignments) and HMM structures
  – in addition: LDA (acoustic context), phonetic decison trees (CART), speaker adaptation, ...

Lessons Learned from Pattern Recognition

- **learning from examples:**
  - **statistical estimation and machine learning**
  - **smoothing and unseen events (e.g. trigram language model)**
  - **suitable training criteria (Max.Lik, MMI, MCE, ...)**

- **search (= max operation in Bayes decision rule):**
  - **advantage: consistent and holistic criterion**
  - **avoid local decisions (interaction between 10-ms level and sentence level; no distinction between statistical PR and syntactical/structural PR)**
  - **cost: complexity of search**
  - **experiments: dynamic programming beam search**

Klatt in 1980 about the principles of DRAGON and HARPY (1976);
p. 261/2 in ‘Lea, W. (1980): Trends in Speech Recognition’:

> **“...the application of simple structured models to speech recognition. It might seem to someone versed in the intricacies of phonology and the acoustic-phonetic characteristics of speech that a search of a graph of expected acoustic segments is a naive and foolish technique to use to decode a sentence. In fact such a graph and search strategy (and probably a number of other simple models) can be constructed and made to work very well indeed if the proper acoustic-phonetic details are embodied in the structure”.**

my adaption to statistical MT (Ney 2008):

> **“...the application of simple structured models to machine translation. It might seem to someone versed in the intricacies of morphology and the syntactic-semantic characteristics of language that a search of a graph of expected sentence fragments is a naive and foolish technique to use to translate a sentence. In fact such a graph and search strategy (and probably a number of other simple models) can be constructed and made to work very well indeed if the proper syntactic-semantic details are embodied in the structure”.**

Today's Statistical MT

four key components in building today's MT systems:

- **training:**
  word alignment and probabilistic lexicon of (source,target) word pairs
- **phrase extraction:**
  find (source,target) fragments (='phrases') in bilingual training corpus
- **log-linear model:**
  combine various types of dependencies between F and E
- **generation (search, decoding):**
  generate most likely (='plausible') target sentence

ASR: some similar components (not all!)

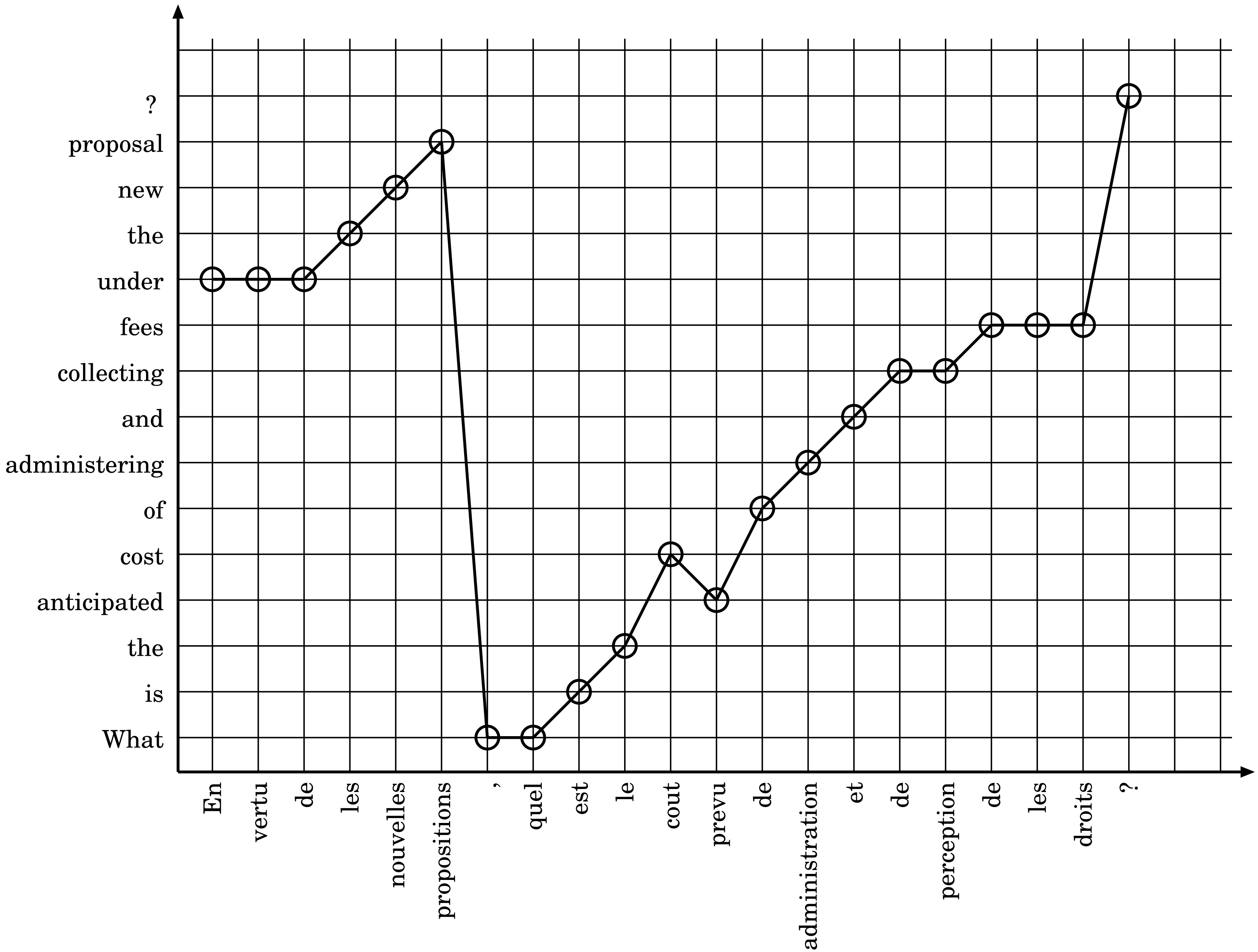
4.2 Training

starting point: probabilistic models in Bayes decision rule:

$$F \rightarrow \hat{E}(F) \;=\; \arg\max_{E} \left\{ p(E|F) \right\} \;=\; \arg\max_{E} \left\{ p(E) \cdot p(F|E) \right\}$$

- **distributions $p(E)$ and $p(F|E)$:**
  – **are unknown and must be learned**
  – **complex: distribution over strings of symbols**
  – **using them directly is not possible (sparse data problem)!**
- **therefore: introduce (simple) structures by decomposition into smaller 'units'**
  – **that are easier to learn**
  – **and hopefully capture some true dependencies in the data**
- **example: ALIGNMENTS of words and positions:**
  bilingual correspondences between words (rather than sentences)
  (counteracts sparse data and supports generalization capabilities)

Example of Alignment (Canadian Hansards)



HMM: Recognition vs. Translation

speech recognition	text translation
$Pr(x_1^T \mid T, w) = \sum_{s_1^T} \prod_t [p(s_t \mid s_{t-1}, S_w, w)\ p(x_t \mid s_t, w)]$	$Pr(f_1^J \mid J, e_1^I) = \sum_{a_1^J} \prod_j [p(a_j \mid a_{j-1}, I)\ p(f_j \mid e_{a_j})]$
time $t = 1, ..., T$ observations x_1^T with acoustic vectors x_t states $s = 1, ..., S_w$ of word w path: $t \rightarrow s = s_t$ always: monotonic	source positions $j = 1, ..., J$ observations f_1^J with source words f_j target positions $i = 1, ..., I$ with target words e_1^I alignment: $j \rightarrow i = a_j$ partially monotonic
transition prob. $p(s_t \mid s_{t-1}, S_w, w)$ emission prob. $p(x_t \mid s_t, w)$	alignment prob. $p(a_j \mid a_{j-1}, I)$ lexicon prob. $p(f_j \mid e_{a_j})$

HMM: first-order dependence in alignments:

$$p(f_1^J, a_1^J | J, e_1^I) \;=\; \prod_j p(a_j | a_{j-1}, J, I) p(f_j | e_{a_j})$$

$$p(f_1^J | J, e_1^I) \;=\; \sum_{a_1^J} p(f_1^J, a_1^J | J, e_1^I)$$

IBM models 1–5 introduced in 1993:

- **IBM-1: = IBM-2 with uniform alignment probabilities**
- **IBM-2: zero-order dependence in alignments**

$$p(f_1^J, a_1^J | J, e_1^I) \;=\; \prod_j p(a_j | j, J, I) p(f_j | e_{a_j})$$

$$p(f_1^J | J, e_1^I) \;=\; \sum_{a_1^J} p(f_1^J, a_1^J | J, e_1^I) = ... = \prod_j \sum_i p(i | j, J, I) p(f_j | e_i)$$

- **IBM-3: = IBM-2 using inverted alignments**
  $i \rightarrow j = b_i$ **with fertility concept**
- **IBM-4: inverted alignment with first-order dependency**
  and dependence of relative distance $j - j'$ and word classes
- **IBM-5: = IBM-4 with proper normalization**

standard procedure:

- **sequence of IBM-1,...,IBM-5 and HMM models: (conferences before 2000; Comp.Ling.2003+2004)**
- **EM algorithm (and its approximations)**
- **implementation in public software (GIZA++)**

remarks on training:

- **based on single word lexica $p(f|e)$ and $p(e|f)$; no context dependency**
- **simplifications: only IBM-1 and HMM**

alternative concept for alignment (and generation):

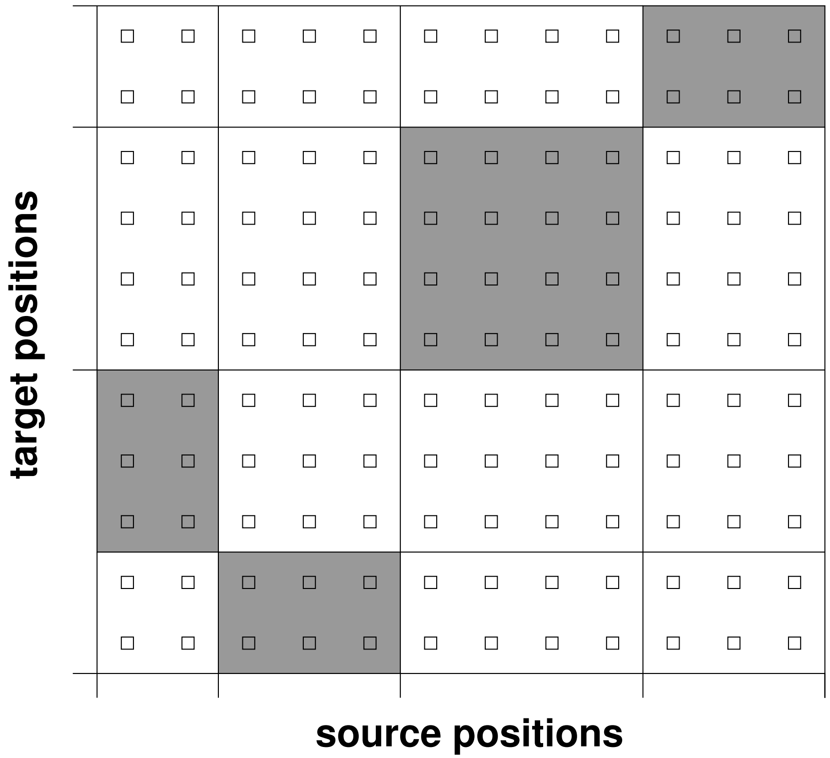
ITG approach [Wu ACL 1995/6]

4.3 Phrase Extraction

segmentation into two-dim. 'blocks'

blocks have to be "consistent" with the word alignment:

- **words within the phrase cannot be aligned to words outside the phrase**
- **unaligned words are attached to adjacent phrases**



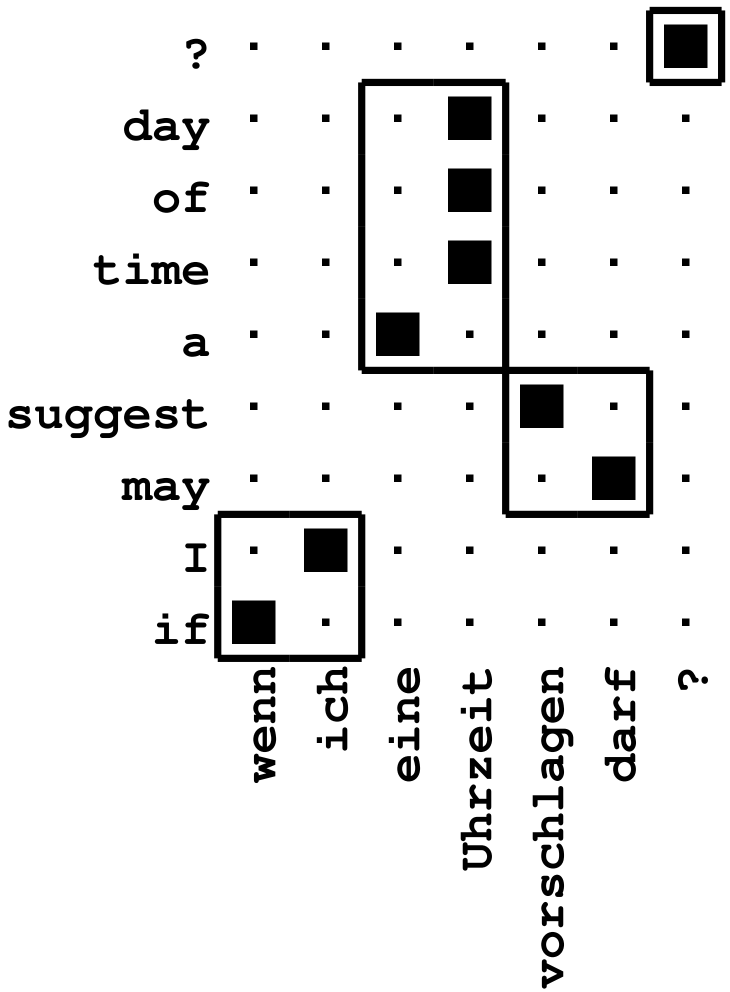
purpose: decomposition of a sentence pair (F, E) into phrase pairs $(\tilde{f}_k, \tilde{e}_k), k = 1, ..., K$:

$$p(E|F) \;=\; p(\tilde{e}_1^K|\tilde{f}_1^K) \;=\; \prod_k p(\tilde{e}_k|\tilde{f}_k)$$

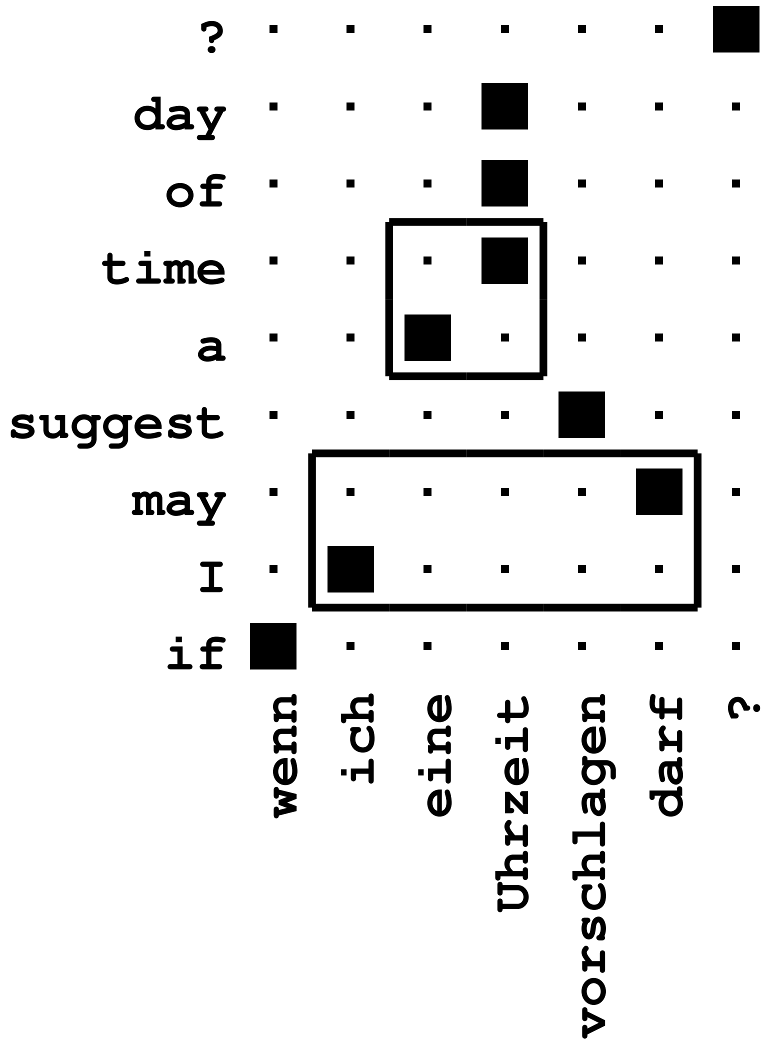
(after suitable re-ordering at phrase level)

Phrase Extraction: Example

possible phrase pairs:



impossible phrase pairs:



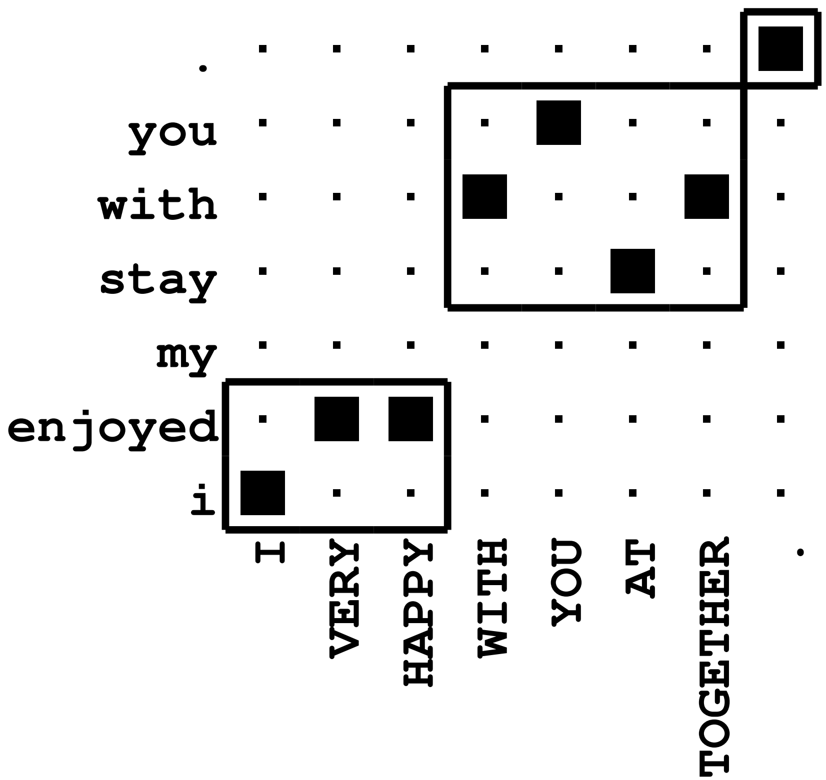
Example: Alignments for Phrase Extraction

source sentence 我 很 高兴 和 你 在 一起 .

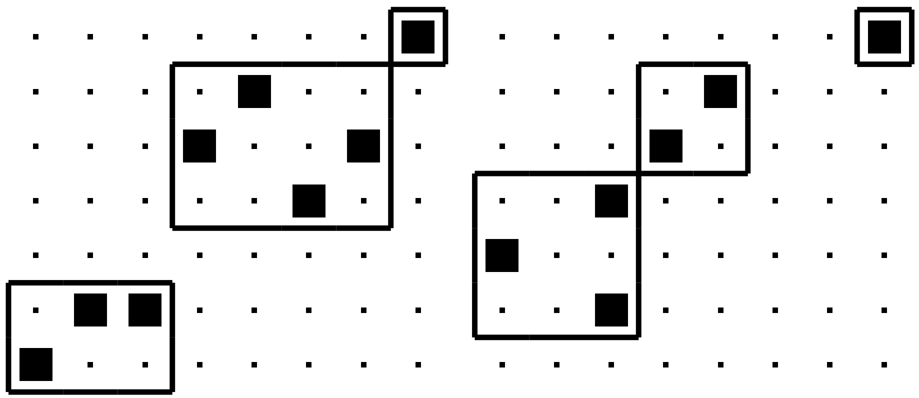
gloss notation I VERY HAPPY WITH YOU AT TOGETHER .

target sentence I enjoyed my stay with you .

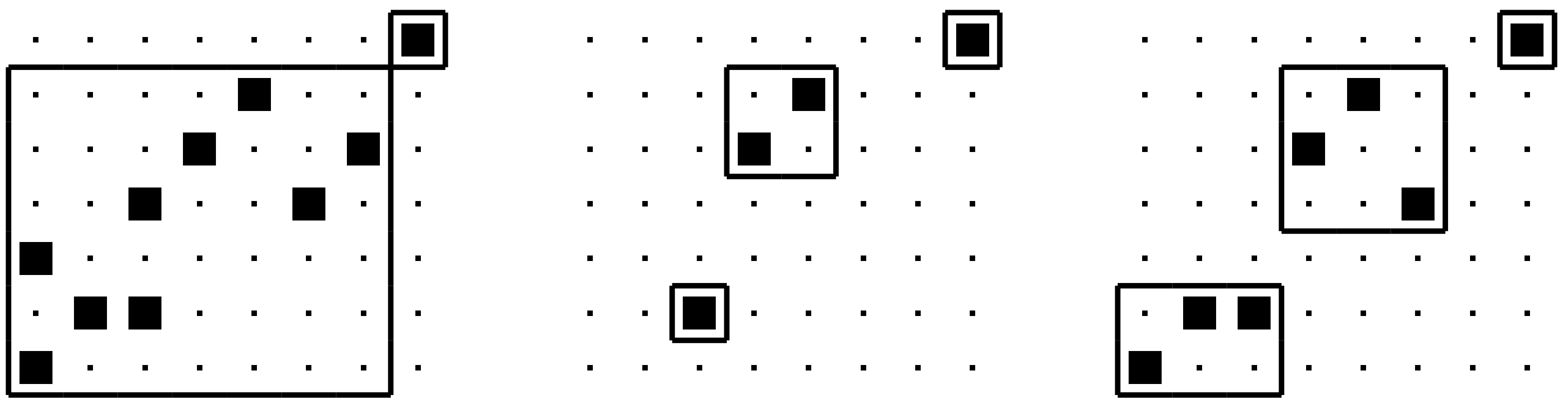
Viterbi alignment for $F \rightarrow E$:



Example: Alignments for Phrase Extraction



Viterbi: $F \rightarrow E$ Viterbi: $E \rightarrow F$



union intersection refined

Alignments for Phrase Extraction

most alignment models are asymmetric:
$F \rightarrow E$ and $E \rightarrow F$ will give different results

in practice: combine both directions using heuristics

- *intersection*: only use alignments where both directions agree
- *union*: use all alignments from both directions
- *refined*: start from *intersection* and include adjacent alignments from each direction

effect on number of extracted phrases and on translation quality (IWSLT 2005)

heuristic	# phrases	BLEU[%]	TER[%]	WER[%]	PER[%]
union	489 035	49.5	36.4	38.9	29.2
refined	1 055 455	54.1	34.9	36.8	28.9
intersection	3 582 891	56.0	34.3	35.7	29.2

4.4 Phrase Models and Log-Linear Scoring

combination of various types of dependencies using log-linear framework (maximum entropy):

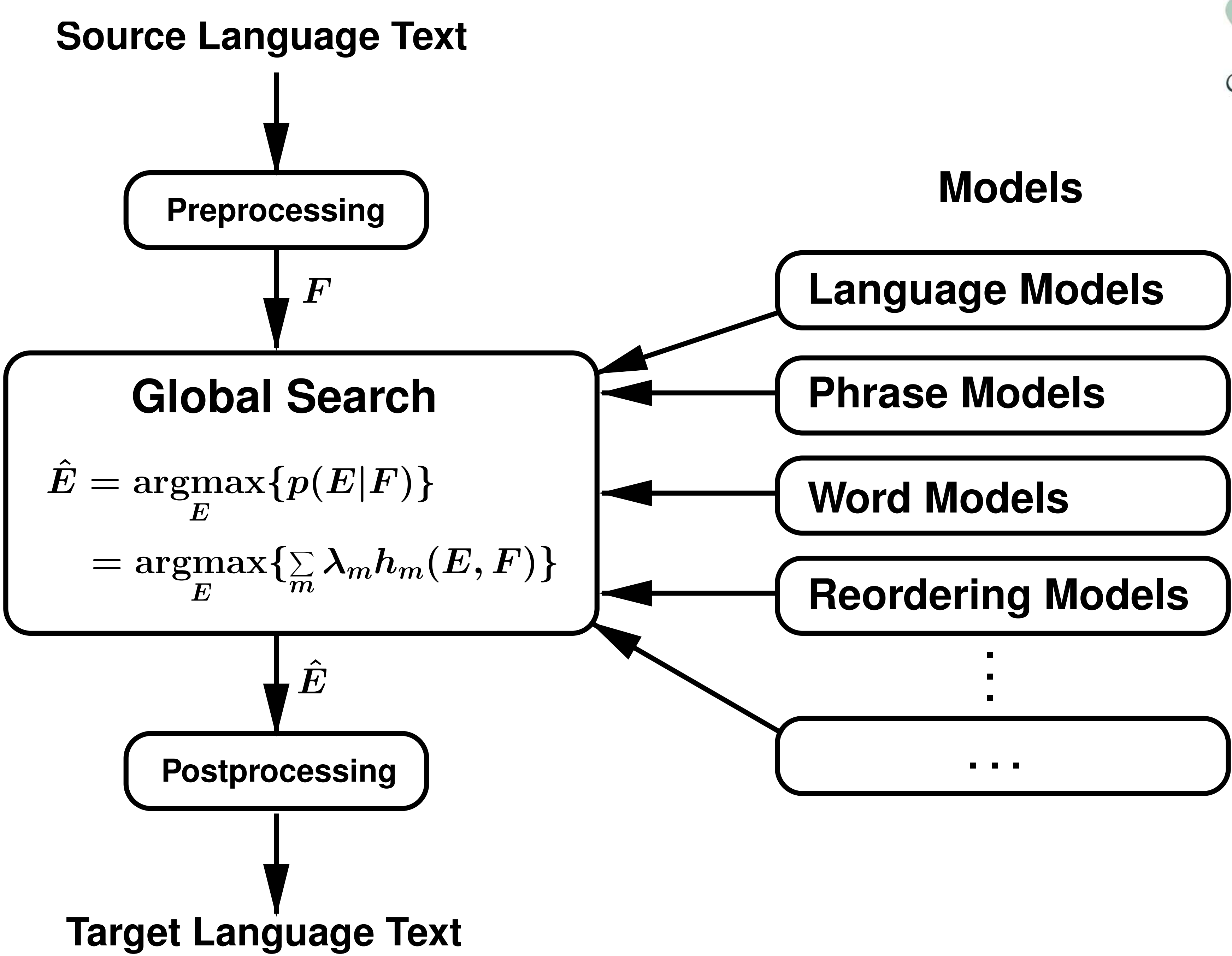
$$p(E|F) \;=\; \frac{\exp\left[\sum_m \lambda_m h_m(E,F)\right]}{\sum_{\tilde{E}} \exp\left[\sum_m \lambda_m h_m(\tilde{E},F)\right]}$$

with 'models' (feature functions) $h_m(E,F), m = 1, ..., M$

Bayes decision rule:

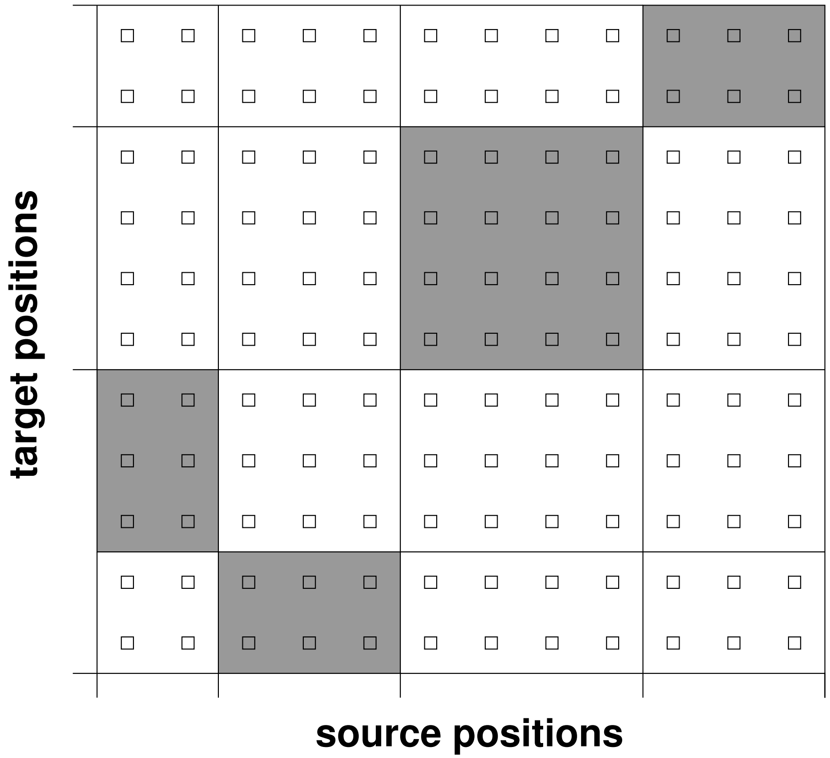
$$F \;\rightarrow\; \hat{E}(F) \;=\; \operatorname*{argmax}_E \left\{ p(E|F) \right\} \;=\; \operatorname*{argmax}_E \left\{ \exp\left[\sum_m \lambda_m h_m(E,F)\right]\right\}$$
$$=\; \operatorname*{argmax}_E \left\{ \sum_m \lambda_m h_m(E,F) \right\}$$

consequence:
– do not worry about normalization
– include additional 'feature functions' by checking BLEU ('trial and error')


Source Language Text
Preprocessing
F
Global Search
$\hat{E} = \operatorname*{argmax}_{E}\{p(E|F)\}$
$= \operatorname*{argmax}_{E}\{\sum_{m} \lambda_m h_m(E,F)\}$
$\hat{E}$
Postprocessing
Target Language Text
Models
Language Models
Phrase Models
Word Models
Reordering Models
...

Phrase Model Scoring

most models $h_m(E, F)$ are based on segmentation into two-dim. 'blocks' $k := 1, ..., K$



five baseline models:

- **phrase lexicon in both directions:**
  – $p(\tilde{f}_k|\tilde{e}_k)$ and $p(\tilde{e}_k|\tilde{f}_k)$
  – estimation: relative frequencies
- **single-word lexicon in both directions:**
  – $p(f_j|\tilde{e}_k)$ and $p(e_i|\tilde{f}_k)$
  – model: IBM-1 across phrase
  – estimation: relative frequencies
- **monolingual (fourgram) LM**

7 free parameters: 5 exponents + phrase/word penalty

history:

- Och et al.; EMNLP 1999:
  – alignment templates ('with alignment information')
  – and comparison with single-word based approach

- Zens et al., 2002: German Conference on AI, Springer 2002;
  phrase models used by many groups
  (Och $\rightarrow$ ISI, Google, ...)

later extensions,
mainly for rescoring N-best lists:

- phrase count model
- IBM-1 $p(f_j|e_1^I)$
- deletion model
- word n-gram posteriors
- sentence length posterior

Experimental Results: Chin-Engl. NIST

		BLEU[%]	
Search	Model	Dev	Test
monotonic	4-gram LM + phrase model $p(\tilde{f}\|\tilde{e})$	31.9	29.5
	+ word penalty	32.0	30.7
	+ inverse phrase model $p(\tilde{e}\|\tilde{f})$	33.4	31.4
	+ phrase penalty	34.0	31.6
	+ inverse word model $p(e\|\tilde{f})$ (noisy-or)	35.4	33.8
non-monotonic	+ distance-based reordering	37.6	35.6
	+ phrase orientation model	38.8	37.3
	+ 6-gram LM (instead of 4-gram)	39.2	37.8

Dev: NIST'02 eval set; Test: combined NIST'03-NIST'05 eval sets

Re-ordering Models

soft constraints ('scores'):

- **distance-based reordering model**
- **phrase orientation model**

hard constraints (to reduce search complexity):

- **level of source words:**
  - **local re-ordering**
  - **IBM (forward) constraints**
  - **IBM backward constraints**
- **level of source phrases:**
  - **IBM constraints (e.g. #skip=2)**
  - **side track: ITG constraints**

Re-ordering Constraints

dependence on specific language pairs:

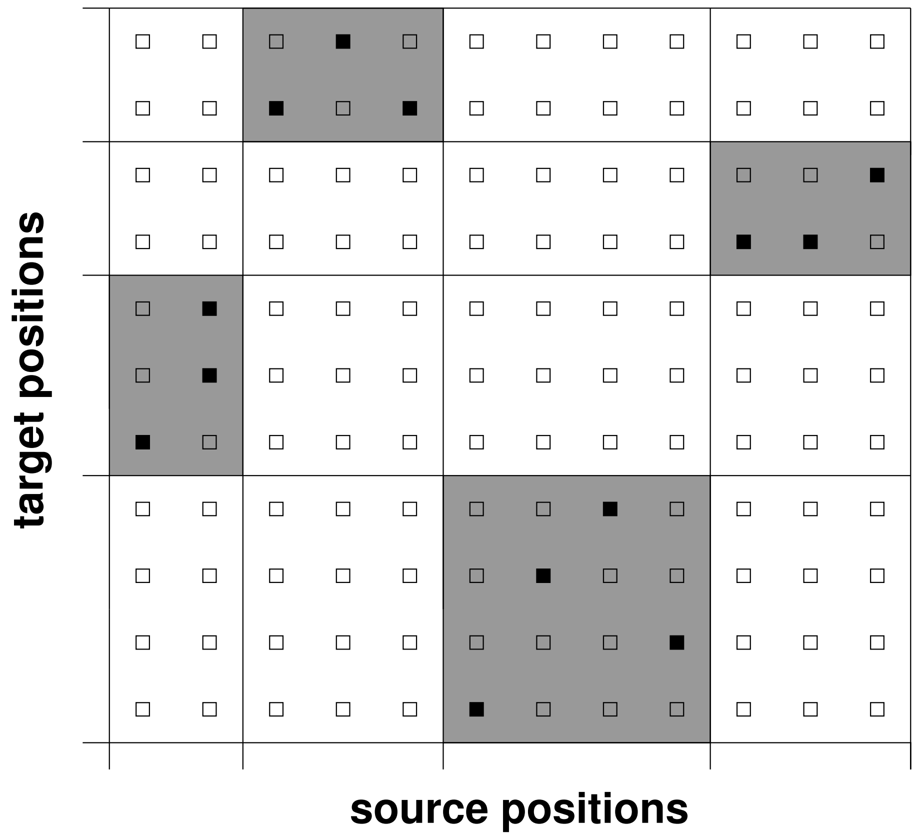
- **German - English**
- **Spanish - English**
- **French - English**
- **Japanese - English**
- **Chinese - English**
- **Arabic - English**

4.5 Generation

constraints:
no empty phrases, no gaps
and no overlaps

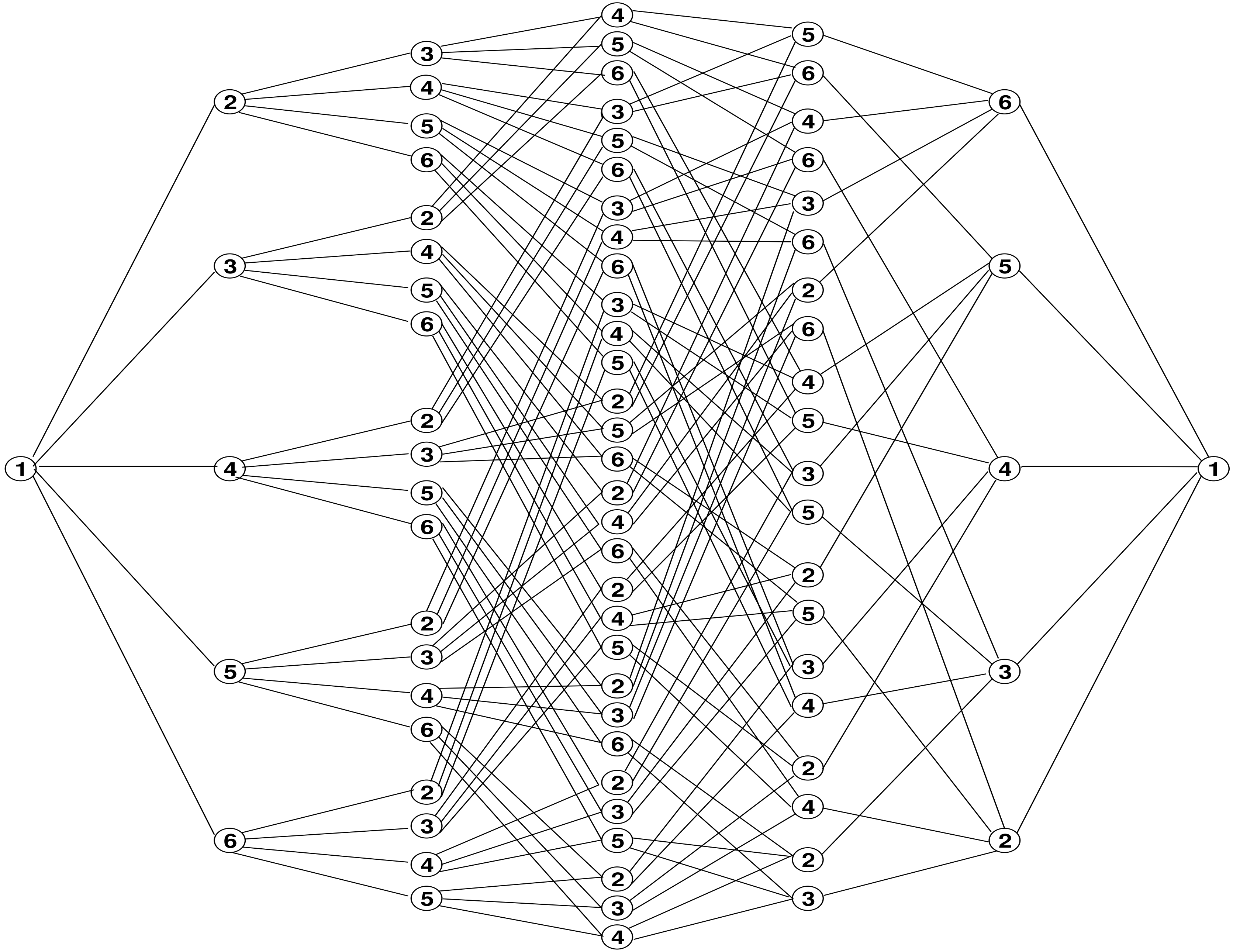
operations with interdependencies:
– find segment boundaries
– allow re-ordering in target language
– find most 'plausible' sentence

similar to: memory-based and
example-based translation



search strategies:
(Tillmann et al.: Coling 2000, Comp.Ling. 2003; Ueffing et al. EMNLP 2002)

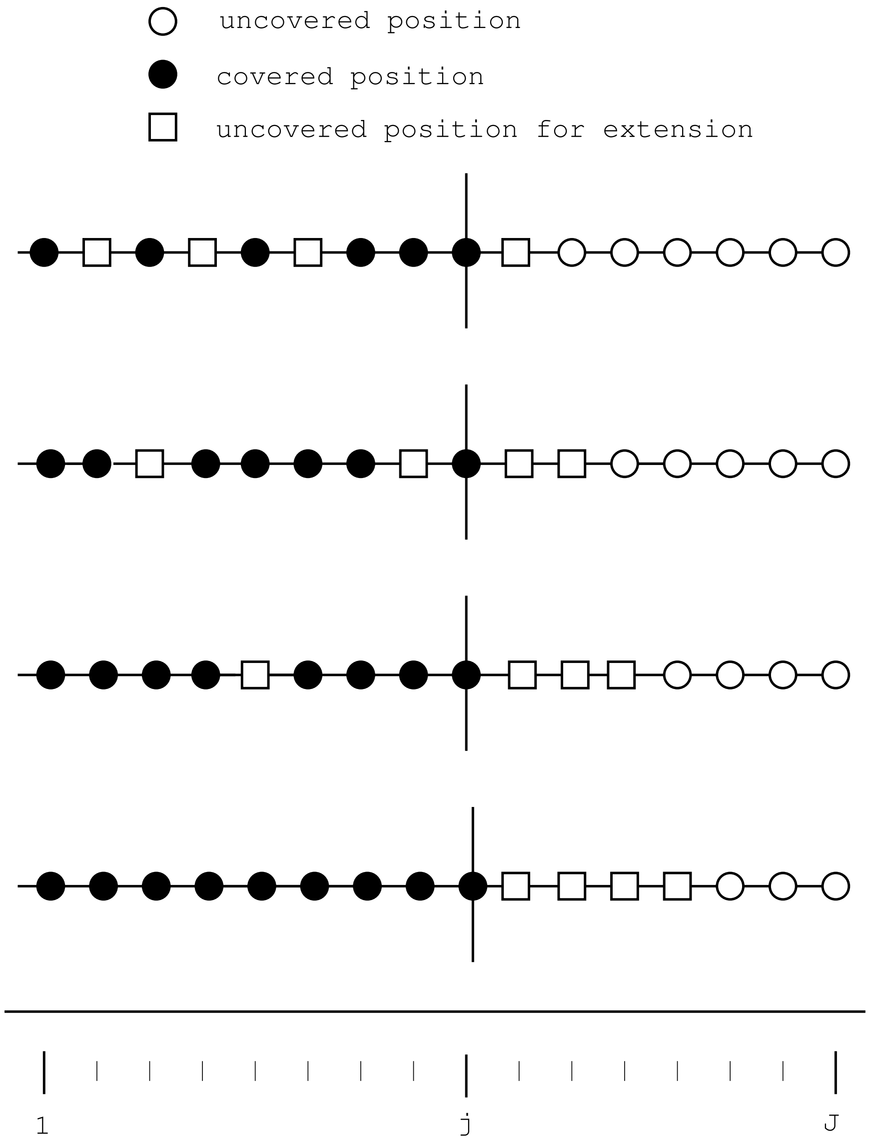
Travelling Salesman Problem: Redraw Network (J=6)



Reordering: IBM Constraints

IBM constraints:
'#skip=3'

result: limited reordering lattice



extensions:
– phrases rather than words
– rest cost estimate for uncovered positions

input: source language string $f_1 ... f_j ... f_J$

for each cardinality $c = 1, 2, ..., J$ do

- **for each set $C \subset \{1, ..., J\}$ of covered positions with $|C| = c$ do**
  - **for each target suffix string $\tilde{e}$ do**
    - – evaluate score $Q(C, \tilde{e}) := ...$
    - – apply beam pruning

traceback:

- – recover optimal word sequence

dynamic programming beam search:

- **build up hypotheses of increasing cardinality:**
  each hypothesis $(C, \tilde{e})$ has two parts:
  coverage hyp. (C) + lexical hyp. $(\tilde{e})$
- **consider and prune competing hypotheses:**
  – with the same coverage vector
  – with the same cardinality

4.6 Summary

today's statistical MT:

- **word alignment (IBM,HMM): learning from bilingual data**
- **from words to phrases:**
  phrase extraction, scoring models and generation (search) algorithms
- **experience with various tasks and 'distant' language pairs:**
  better than rule-based approaches
- **text + speech**

room for improvements:

- **training of phrase models:**
  right now: more extraction than training
- **improved alignment and lexicon models:**
  more complex models in lieu of $p(e|f)$
- **phrase and word re-ordering:**
  – **long-distance dependencies**
  – **hierarchical ('gappy') phrases [Chiang 2005]**
  – **syntax [Marcu et al. 2006]**

5 Image Recognition

interest: strictly appearance-based approach:

- **appearance based concept,**
  i.e. no explicit extraction of features
- **avoid segmentation:**
  interdependence between object recognition and boundary detection
- **matching: each pixel of the test image must be matched**
  against a pixel in the reference image
- **pixel representation: grey level + neigbourhood ('derivatives')**

contrast: more conventional approach:

- **decomposition of image into patches and**
  extraction of features and descriptors (SIFT)
- **classifier: Gaussian mixture**

competitive results on CalTech database and in PASCAL evaluations;
papers by Deselaers et al.

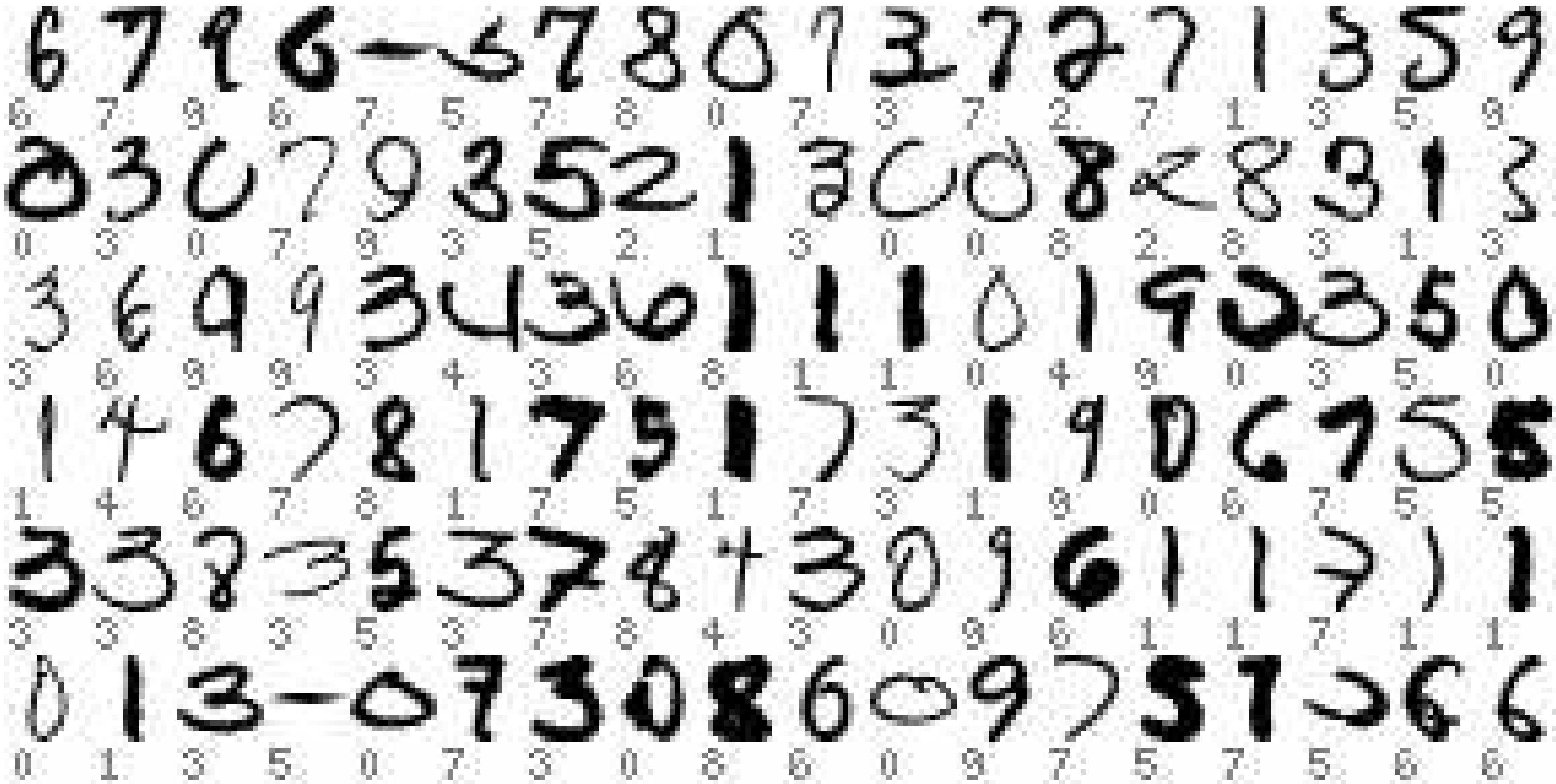
ingredients of appearance-based approach:

- **observations: Gaussian distribution for pixel vectors**
- **matching: alignment model:**
$$t = (i, j) \rightarrow s_t = (u, v)_{ij}$$
position: vertical and horizontal coordinates
- **problem: first-order dependencies require Markov random field (MRF)**
$\rightarrow$ exponential complexity for sum in denominator

possible approximations:

- **convert the problem into a 1-D problem:**
– approprate for continuous cursive handwriting
– similar to ASR approach in virtually all details
– competitive results on Arabic and English tasks [ICDAR 2009, Dreuw et al.]
- **zero-order model for alignments: $p(s_t|t, c)$ in lieu of $p(s_t|s_{t-1}, c)$**
– appropriate for digits: USPS (16^2 pixels) and MNIST (26^2 pixels) database
– advantage: polynomial complexity for sum in denominator

USPS: Examples



method for handwritten digits (USPS and MNIST):

- **appearance based concept,**
  i.e. no explicit extraction of features
- **pixel: grey level, various derivatives**
- **zero-order alignment model ('IDM: image distortion model')**
- **Gaussian model with discriminative training: log-linear model**

extensions towards face recognition

experimental results:

- **distortion model is important**
- **competitve results**
  with comparatively small models

Experiments on MNIST and USPS

	MNIST		USPS	
Model	# param.	ER	# param.	ER
NN	47,040,000	3.1%	1,866,496	5.6%
NN + IDM$^{(1)}$	47,040,000	0.6%	1,866,496	2.4%
single Gaussians	7,840	18.0%	2,560	18.5%
single Gaussians + IDM$^{(1)}$	7,840	5.8%	2,560	6.5%
SVM	?	1.5%	532,000	4.4%
SVM + IDM $^{(1)}$	-	-	532,000	2.8%
log-linear model:				
grey values (no IDM)	7,850	7.4%	2,570	8.5%
{derivatives} + IDM	227,370	1.3%	69,130	3.5%
{derivatives} + IDM + tying	31,390	1.3%	5,220	3.8%
deep belief network	1,665,010	1.3%	640,610	-
conv. network	2,406,325	0.4%	-	-

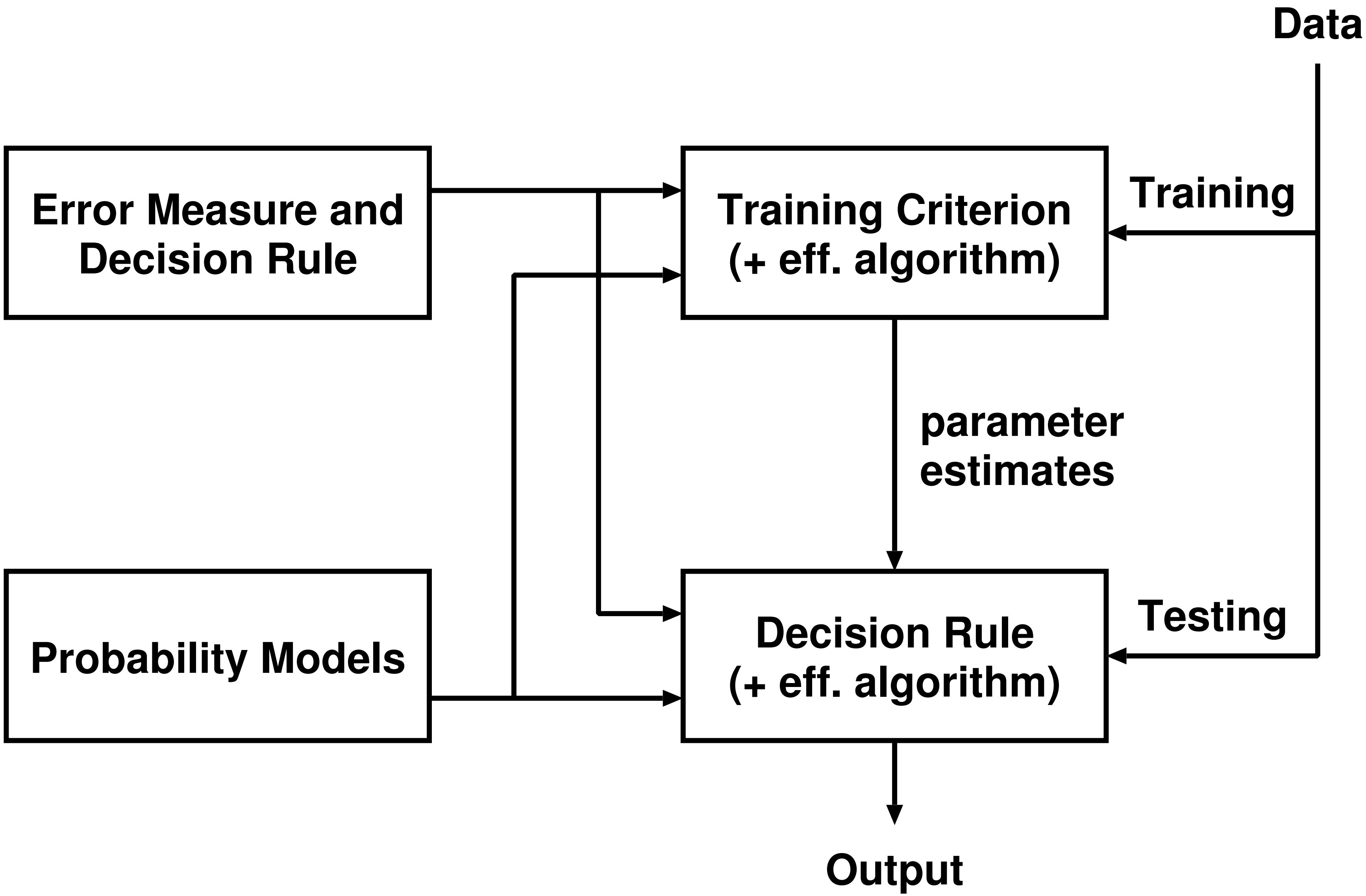
$^{(1)}$ additional features in distance computation

open questions:

- **MRF: suitable approximations ('belief propagation')**
  (for either sum or maximum)
- **feature extraction:**
  result of discriminative training
- **more challenging tasks**
  and more powerful algorithms for matching

6 Conclusion

four key ingredients for ASR, MT and image recognition:



THE END