Red Temàtica en Tecnologías del Habla (RTTH)
RTTH Fall School 2023, Nov. 14-17, 2023

Jaca, Aragon, Spain

Keynote, 14-Nov-2023:

Data-Driven Speech & Language Technology (HLT): from Small to Large Models

Hermann Ney

RWTH Aachen University, Aachen, Germany
AppTek, Aachen, Germany & McLean, VA

This Lecture

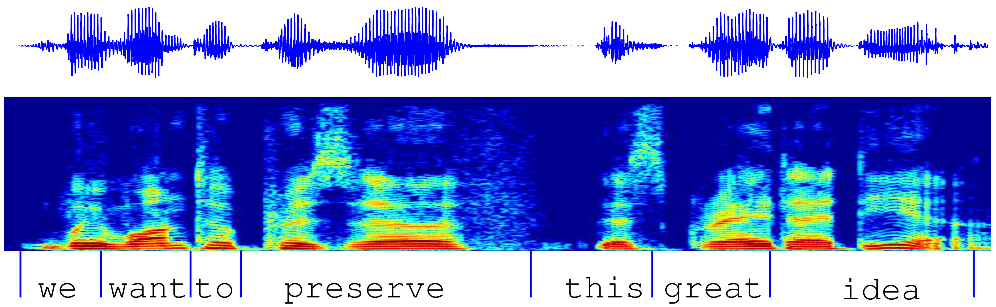
- **my personal interpretation (experience: 1978-2023):**
  - **unifying framework: probabilistic models and Bayes decison theory**
  - **deep learning is just one out of many machine learning approaches**
  - **experience: 'more data help'**
- **messages:**
  - **success of data-driven approaches**
  - **NLP and AI: moving from rule-based to data-driven approaches**
  - **things started 40 years ago, not in 2013!**
  - **evolution from small to large language (and acoustic!) models**
  - **sort out the fundamental principles beyond experimental noise**
  - **framework: (applied) mathematical and statistics**
- **key messages:**
  - **there has been, is and will be life outside deep learning**
  - **there is NO life outside probabilistic modelling (Bayes framework)**

Outline

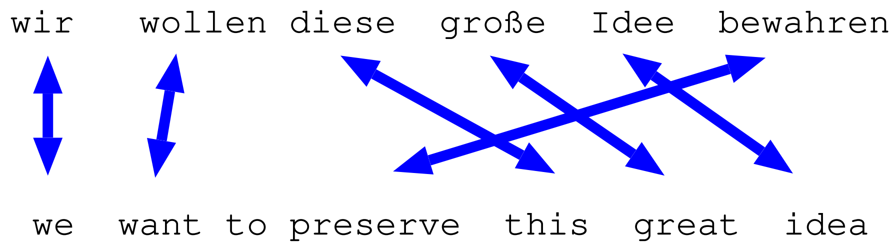
1 HLT and ANNs

Speech & Language Technology: Sequence-to-Sequence Processing

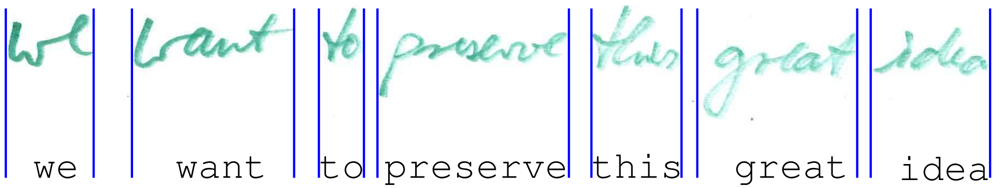
Automatic Speech Recognition (ASR)
(speech signal processing)



Machine Translation (MT)
(symbol or text processing)



Handwriting Recognition (HWR)
(text image processing)



common characteristics:

- **use of a 'small' language model (LM) to generate smooth fluent text (syntax, semantics, context)**
- ***generative* aspect of LM: unlike *formal* NLP tasks (POS/synt./semant. labels, ...)**
- **LM is learned from text only (*without annotation, unsup. mode, pre-training)***

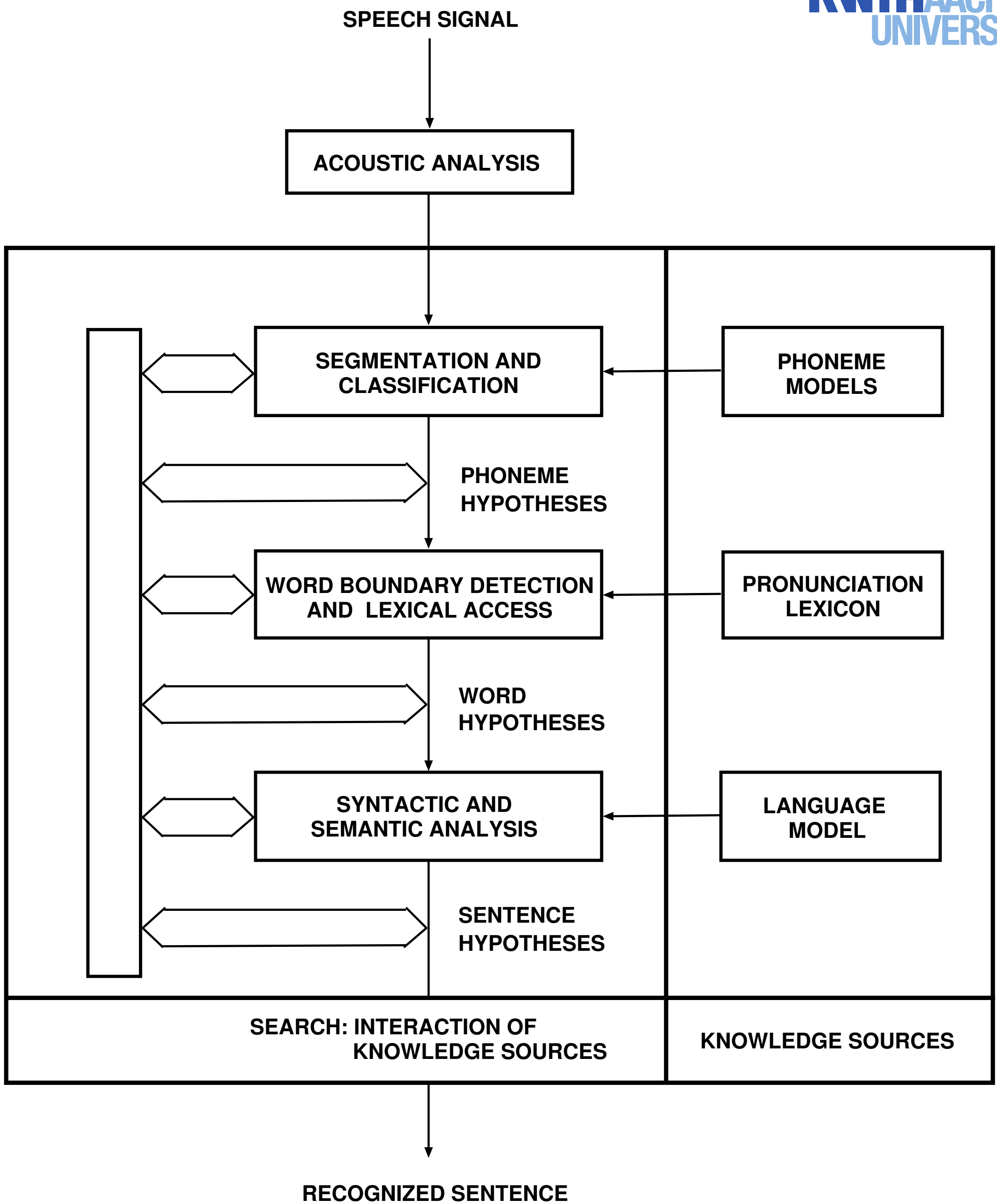
note: this is how (small) language models started (1980 - 2000)
[Jelinek & Mercer^{+} 77]

ASR: first research 1975-1980

ASR is sequence-to-sequence processing at several levels:
10-ms vectors, phonemes, words

problems:
- **ambiguities at/between all levels**
- **interdependencies of decisions**

approach 1975-1980 (Baker/CMU and Jelinek/IBM):
- **probabilitistic modelling**
- **holistic approach ('end-to-end'): single criterion for system design (Bayes decision rule)**
- **complex mathematical modelling**



1980-1995 Baseline Statistical Approach to ASR

- **modelling: probability distributions/data-driven approaches with**

$$\textbf{10-msec vectors:} \quad x_1^T = x_1 ... x_t ... x_T \qquad x_t \in \mathbb{R}^D$$
$$\textbf{word string:} \quad w_1^N = w_1 ... w_n ... w_N$$

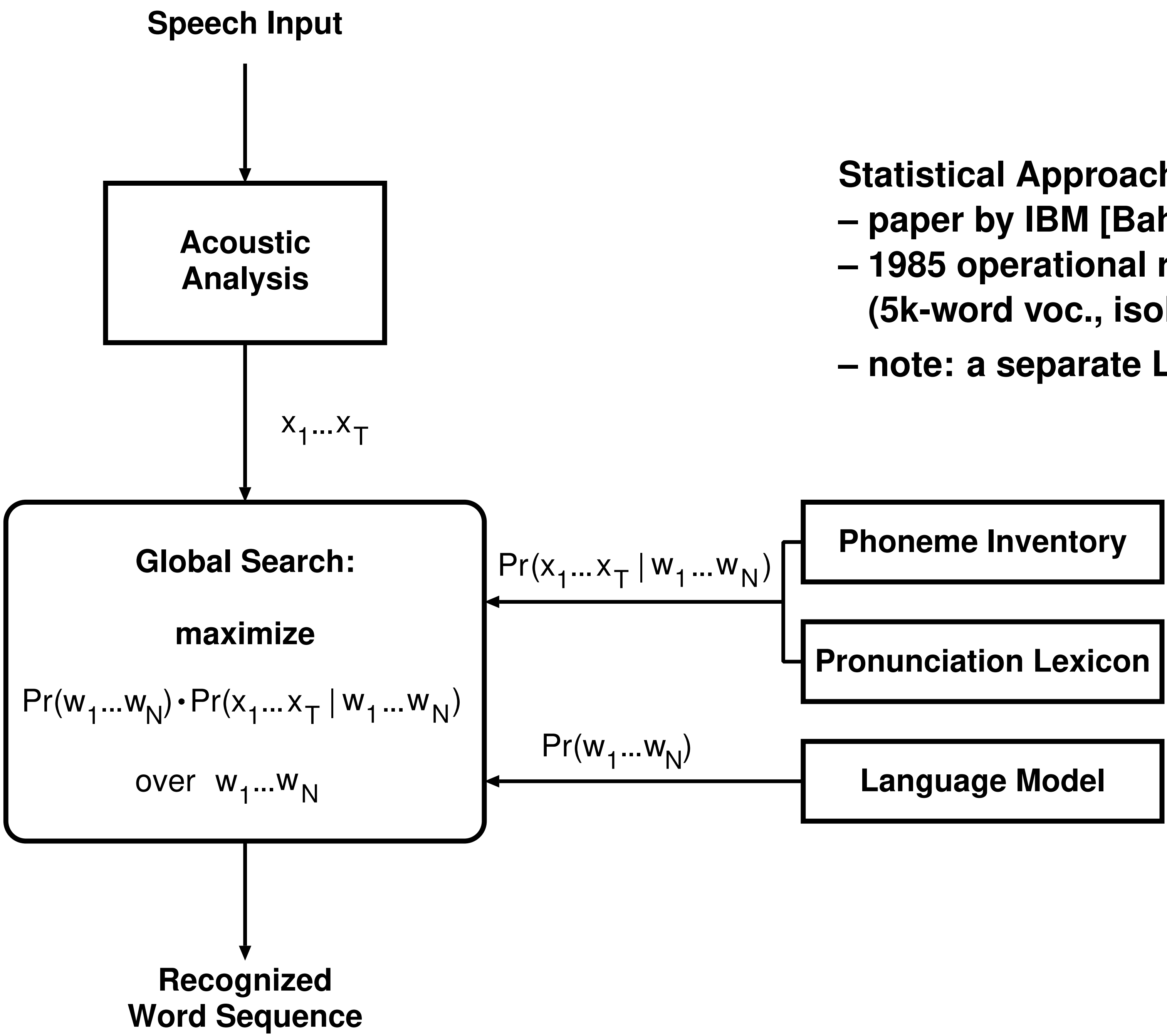
- **consider joint generative model:** $p(w_1^N, x_1^T) = p(w_1^N) \cdot p(x_1^T | w_1^N)$
- **language model $p(w_1^N)$: based on word trigram counts, learned from text only $[w_1^N]$**
- **acoustic (-phonetic) model $p(x_1^T | w_1^N)$: learned from annotated audio data $[x_1^T, w_1^N]$**
  - **generative hidden Markov model:**
    discrete models/VQ, Gaussians, Gaussian mixtures, ...
  - **structure: first-order dependence and mathematically nice**
  - **training: ('efficient') EM algorithm with sort of closed-form solutions**
- **dichotomy:**
  - **general machine learning (like CV): single (isolated) events (x, c):**
    emphasis on 'discriminative' class posterior $p(c|x)$ (rather than $p(x, c) = p(c) \cdot p(x|c)$)
  - **sequence-to-sequence task (like ASR: time alignment and LM context):**
    emphasis on 'generative' joint model $p(x_1^T, w_1^N)$
- **decoding/generation: Bayes decision rule (simplified form)**
  = use single criterion and avoid local decisions

Statistical Approach to ASR

– paper by IBM [Bahl & Jelinek^{+} 83]

– 1985 operational research system: *Tangora*
  (5k-word voc., isolated words, speaker dep.)

– note: a separate LM



Operational ASR Systems

ASR at Philips: Research Hamburg/Aachen and BU Dictation Systems Vienna:

- **1k-word continuous speech recognition: research prototype**
  SPICOS 1984-1989 (German BMBF): Siemens, Philips, German universities

- **10k-word continuous speech recognition: commercial Philips product**
  – speaker dep., DP beam search and dynamic search space, real-time on Motorola 68020
  – presentation at Eurospeech 1993: medical text dictation

speech translation (= ASR + MT) at RWTH Aachen: research prototypes

- **Verbmobil 1993-2000 (German BMBF):**
  appointment scheduling/limited domain, German-English, 8k words

- **TC-STAR 2004-2007: domain: speeches given in EU parliament**
  – challenge: MT robust wrt ASR errors $\rightarrow$ data-driven methods
  – approach to MT: phrase-based approach

  – first research prototype for unlimited domain and real-life data
  ◦ fully automatic, not real time
  ◦ without deep learning!
  – partners: KIT Karlsruhe, RWTH, CNRS Paris, UPC Barcelona, IBM-US Research, ...

more research prototypes: GALE, BOLT, BABEL, QUAERO, EU-Bridge, Translectures, ERC
along with DARPA/NIST/project evaluations

ASR History: Operational Research Systems

- **steady improvement of data-driven methods:**
  HMMs with Gaussians and mixtures, phonetic CART, statistical trigram language model, speaker adaptation, sequence discriminative training, ANNs
- **methodology in ASR since 1990: standard public data:**
  TIMIT, RM/1k, WSJ/5k, WSJ/20k, NAB/64k, Switchboard/tel., Librispeech, TED-Lium
- **1993-2000 NIST/DARPA: comparative evaluation of operational systems:**
  – virtually all systems: generative HMMs and refinements
  – 1994 Robinson: hybrid HMM with RNN (singularity!)

alternative concepts (with less success):

- **1985-93: criticism about data-driven approach/machine learning**
  – acoustic model: too many parameters and saturation effect
  – concept of rule-based AI: acoustic-phonetic expert systems
  – language model: similar criticism (linguistic structures/grammars)
- **SVM (support vector machines): never competitive in ASR**
  (ASR requires decisions in context!)

ASR: ANN in Acoustic Modelling

- **1987 [Bourlard & Wellekens 87]: MLP and ASR**
- **1988 [Waibel & Hanazawa^{+} 88]: phoneme recognition by TDNN (convol.NNs!)**
- **1989 [Bourlard & Wellekens 89, Morgan & Bourlard 90]:**
  - **ANN outputs: can be interpreted as class posteriors**
  - ***hybrid HMM*: use ANN for frame label posteriors**
- **1989 [Bridle 89]: softmax ('Gaussian posterior') for normalized ANN outputs**
- **1991 [Bridle & Dodd 91] backpropagation for HMM discriminative training at word level**
- **1993 [Haffner 93]: sum over label-sequence posterior probabilities in hybrid HMMs (*sequence discriminative training*)**
- **1994 [Robinson 94]: RNN in hybrid HMM (operational system, DARPA evaluations)**
- **1997 [Fontaine & Ris^{+} 97, Hermansky & Ellis^{+} 00]: *tandem HMM*: use ANN for feature extraction in a Gaussian HMM**
- **2009 Graves: CTC for handwriting recognition (operational system, ICDAR competition 2009)**

hybrid HMM: ANN-based feature extraction + Gaussian posterior + HMM

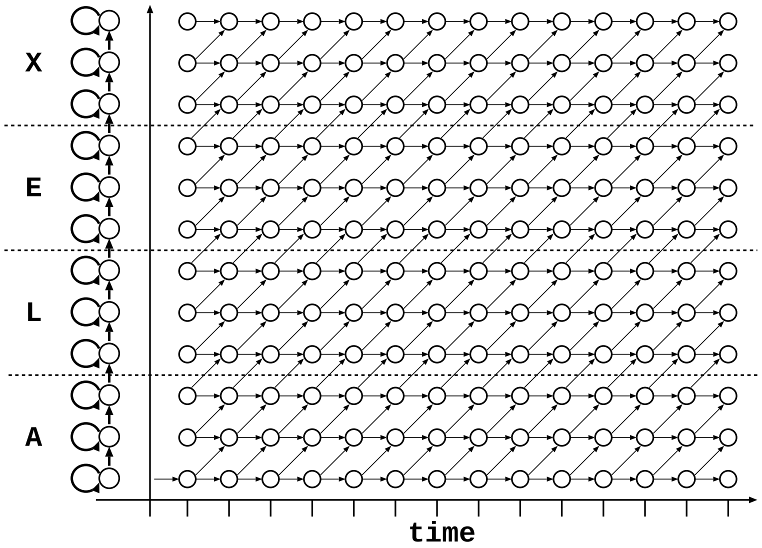
- **2009 [Graves 09]: CTC - good results on LSTM RNN for handwriting task**
- **2010 [Dahl & Ranzato$^+$ 10]: improvement in phone recognition on TIMIT**
- **2011 [Seide & Li$^+$ 11, Dahl & Yu$^+$ 12]: Microsoft Research**
  - **fully-fledged hybrid HMM**
  - **30% rel. WER reduction on Switchboard 300h**
- **since 2012: other teams confirmed reductions of WER by 20% to 30%**

tandem HMM: ANN-based feature extraction + generative Gaussian + HMM

- **2006 [Stolcke & Grezl$^+$ 06]: cross-domain and cross-language portability**
- **2007 [Valente & Vepa$^+$ 07]: 8% rel. WER reduction on LVCSR**
- **2011 [Tüske & Plahl$^+$ 11]: 22% rel. WER reduction on LVCSR/QUAERO (Interspeech 2011, like [Seide & Li$^+$ 11])**

experimental observation for hybrid and tandem HMM:
progress by using *deep* MLPs

Hidden Markov Model (HMM): Classical vs. Hybrid HMM



– **sequence of acoustic vectors:**
$X = x_1^T = x_1 ... x_t ... x_T$ **over time** $t = 1, ..., T$

– **sequence of states/segments** $s = 1, ..., S$
$s_1^T = s_1 ... s_t ... s_T$ **over time** t
with phonetic/graphemic labels:
$a_1^S = a_1 ... a_s ... a_S$
$= W$**: word sequence**

- **classical HMM: generative model for input sequence** x_1^T**:**

$$p(x_1^T | W = a_1^S) = \sum_{s_1^T} \prod_t p(s_{t+1} | s_t, a_{s_t}) \cdot p(x_t | a_{s=s_t})$$

- **hybrid HMM: discriminative model for output sequence** a_1^S**:**
[Bourlard & Wellekens 89] machine learning point-of-view:
it is much(!) better to model $p(a_s | x_t)$ **than** $p(x_t | a_s)$ **:**

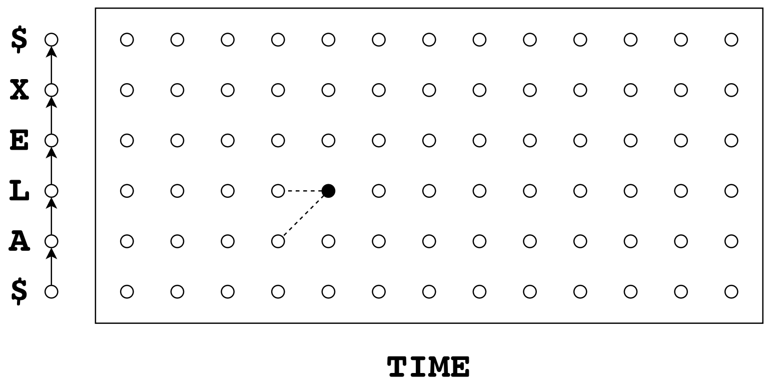
$$p(x_t | a_s) = q(a_s | x_t) \cdot p(x_t) \Big/ q(a_s) \qquad \text{(note: approximative relation!)}$$

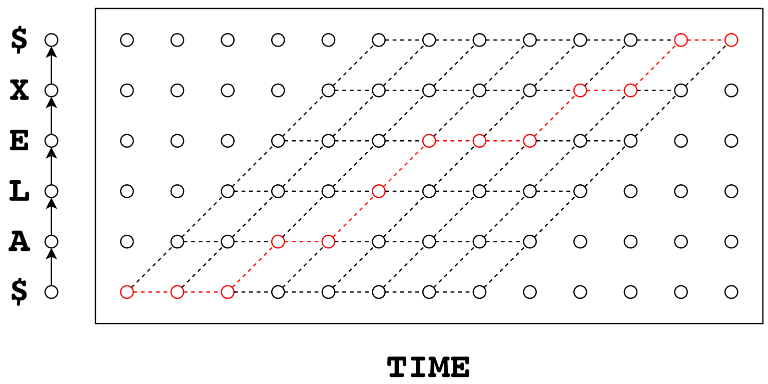
$$p(W = a_1^S | x_1^T) = \sum_{s_1^T} \prod_t p(s_{t+1} | s_t, a_{s_t}) \cdot p(a_{s=s_t} | x_t)$$

Direct or Posterior HMM (Variants: CTC, Transducer, ...) ((view: how to reach $[t,\ s = s_t]$?)

three sequences over time:

$$
\begin{aligned}
x_1^T &= x_1, \ldots, x_t, \ldots, x_T \\
s_1^T &= s_1, \ldots, s_t, \ldots, s_T \\
y_1^T &= y_1, \ldots, y_t, \ldots, y_T
\end{aligned}
$$





path consists of transitions reaching $[t,\ s = s_t]$:
first transition δ_t and then label y_t:

$$[t{-}1,\ s_{t-1}] \rightarrow [t,\ s = s_t = s_{t-1}{+}\delta_t] \qquad \delta_t \in \{0, 1\}$$

JOINT event of δ_t and frame label y_t:

$$[\delta_t, y_t] : \quad p\big([\delta_t,\ y_t]\big|\ldots, x_1^T\big)$$

link to state s with label $a_s \in a_1^S$:

$$[\delta_t, y_t] : \quad p\big([\delta_t,\ y_t = a_s]\big|\ldots, x_1^T\big)$$

first-order dependence in a_1^S:

$$[\delta_t, y_t] : \quad p\big([\delta_t,\ y_t = a_s]\big|a_{s-1}, \ldots, x_1^T\big)$$

remarks:

– **for full context, replace a_{s-1} by a_0^{s-1}**

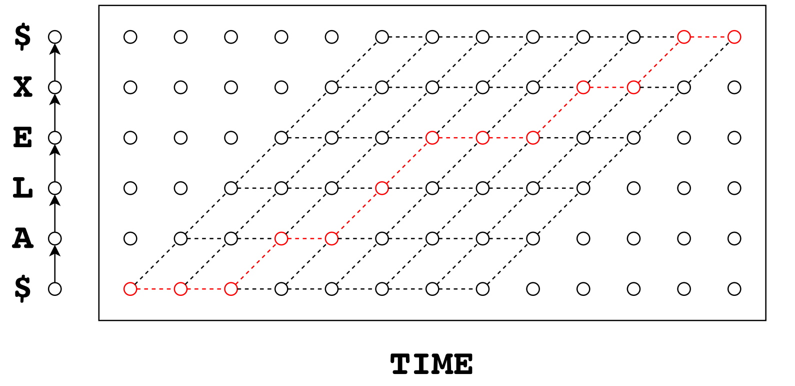
– **alternative view: how to leave $[t,\ s = s_t]$?**
first label y_t and then transition δ_t:

$$p\big([y_t = a_s,\ \delta_t]\big|a_{s-1}, x_1^T\big)$$

Mathematical Formalism:
Direct or Posterior HMM for $p(a_1^S|x_1^T)$ (view: how to reach $[t,\ s = s_t]$?)

RWTH AACHEN UNIVERSITY

formal derivation of full model:



$$p(a_1^S|x_1^T) \;=\; \sum_{s_1^T} p(a_1^S, s_1^T|x_1^T)$$

finite-state model: factorization over t:

first-order model in s_1^T and a_1^S

$$= \sum_{s_1^T} \prod_t p\big([s_t, y_t = a_{s_t}\big| s_{t-1}, a_{s_t-1}, x_1^T\big)$$

difference in state/segment indices: $\delta_t := s_t - s_{t-1}$

$$= \sum_{s_1^T} \prod_t p\big([\delta_t, y_t = a_{s_t}]\big| a_{s_t-1}, x_1^T\big)$$

explicit segmental interpretation:

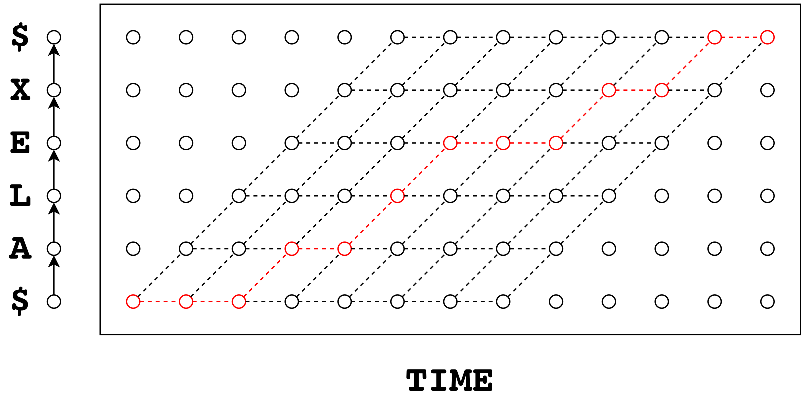
frames t within segment s:
- **first frame: $\delta_t = 1$**
- **other frames: $\delta_t = 0$**

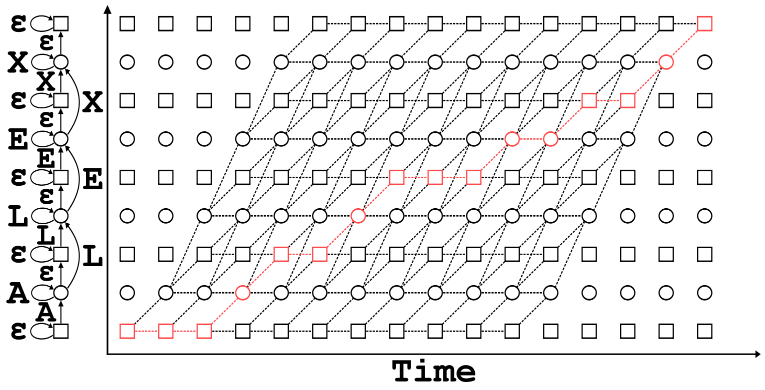
$$= \sum_{s_1^T} \prod_s \prod_{t:\, s_t = s} p\big([\delta_t, y_t = a_s]\big| a_{s-1}, x_1^T\big)$$

acoustic encoder : $h_t = h_t(x_1^T)$

$$= \sum_{s_1^T} \prod_s \prod_{t:\, s_t = s} p\big([\delta_t, y_t = a_s]\big| a_{s-1}, h_t(x_1^T)\big)$$

Direct HMM and Variants:
CTC, [RNN-] Transducer, Blank/ϵ Models





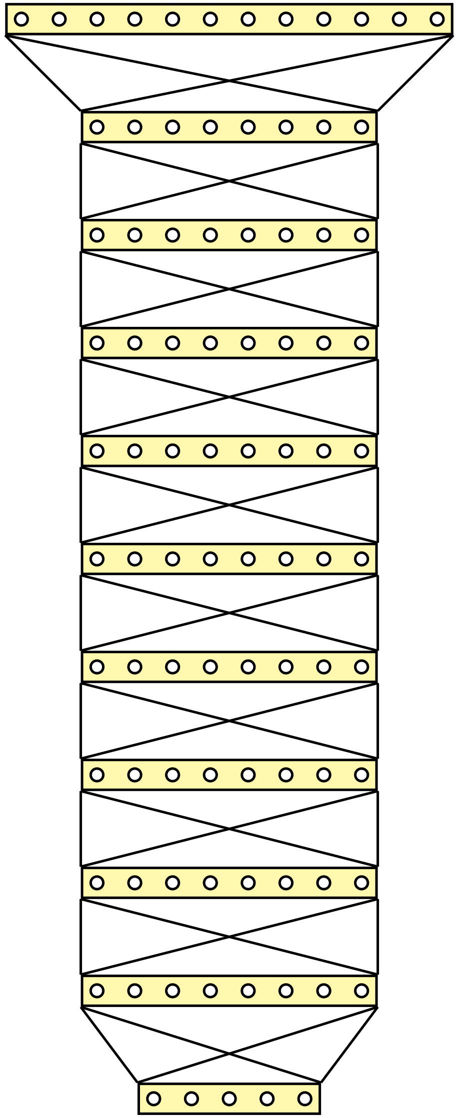
direct HMM: without and without blanks/ϵ

question: how to model the joint event $[\delta_t,\ y_t = a_s]]$ in $p\big([\delta_t, y_t = a_s]\big|a_{s-1}, x_1^T\big)$ **?**
here: no separation of transition and label probabilities !

- **direct HMM (no blanks/ϵ):**
  keep the original joint alphabet for the ANN output nodes:
  $$\Big\{[\delta_t \in \{0,1\},\ y_t = a_s]\Big\}$$ **= 2 x (segment label alphabet) + silence label**

- **transducer: with blanks/ϵ:**
  simplify the alphabet of joint events $[\delta_t,\ y_t = a_s]$:
  $$[\delta_t = 1, y_t = a_s]\ := a_s \qquad\qquad [\delta_t = 0, y_t = a_s]\ := \epsilon$$
  resulting alphabet: 1x (segment label alphabet) + ϵ (also for silence)

Artificial Neural Networks (ANN) and Deep Learning:



question: what is different now after 30 years?

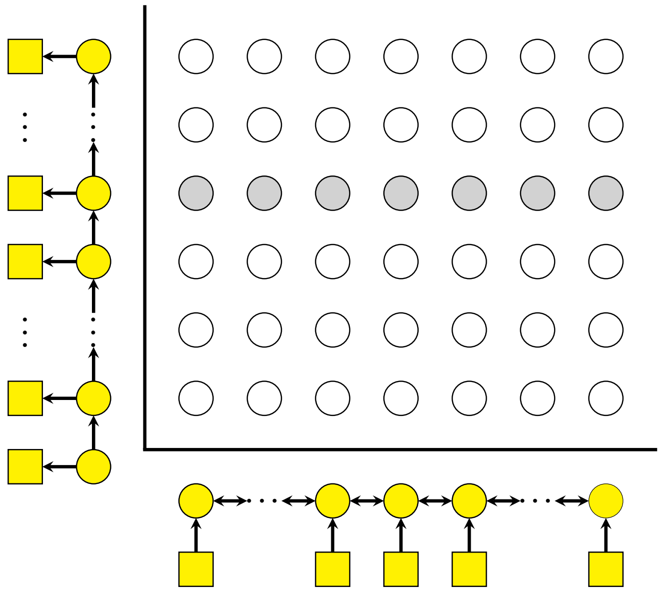
answer: we have learned how to (better) handle a complex numerical optimization problem:

- **more powerful hardware (e. g. GPUs)**
- **empirical recipies for optimization: practical experience and heuristics, e.g. layer-by-layer pretraining**
- **result: we are able to handle more complex architectures (deep MLP, RNN, attention, transformer, etc.)**

my interpretation: 2022's most advanced ASR systems:
$=$ **sophisticated feature extraction/representation**
$+$ **softmax (= Gaussian posterior)**

Input-Output Alignment: Attention and Transducer

common properties:
– input: acoustic encoder: representation/state vectors $h_t = h_t(x_1^T), t = 1, ..., T$
– output: (phoneme) labels $a_s,\ s = 1, ..., S$ with/without integrated language model



representation/state vectors h_t:
– deep MLP: finite window
– RNN and LSTM-RNN
– self-attention (transformer)
similar: output string

- **(cross-) attention: direct factorization:**

$$p(a_1^S|x_1^T) = \prod_s p(a_s|a_0^{s-1}, x_1^T) = \prod_s p(a_s|a_{s-1}, r_{s-1}, c_s)$$

$$c_s := \sum_t p(t|a_0^{s-1}, x_1^T) \cdot h_t$$

with context vector c_s and output state vector r_s

criticism for ASR: lack of strict monotonicity and localization

- **finite-state transducer (direct HMM, CTC, RNN-T, ...): introduce hidden paths and then factorize:**

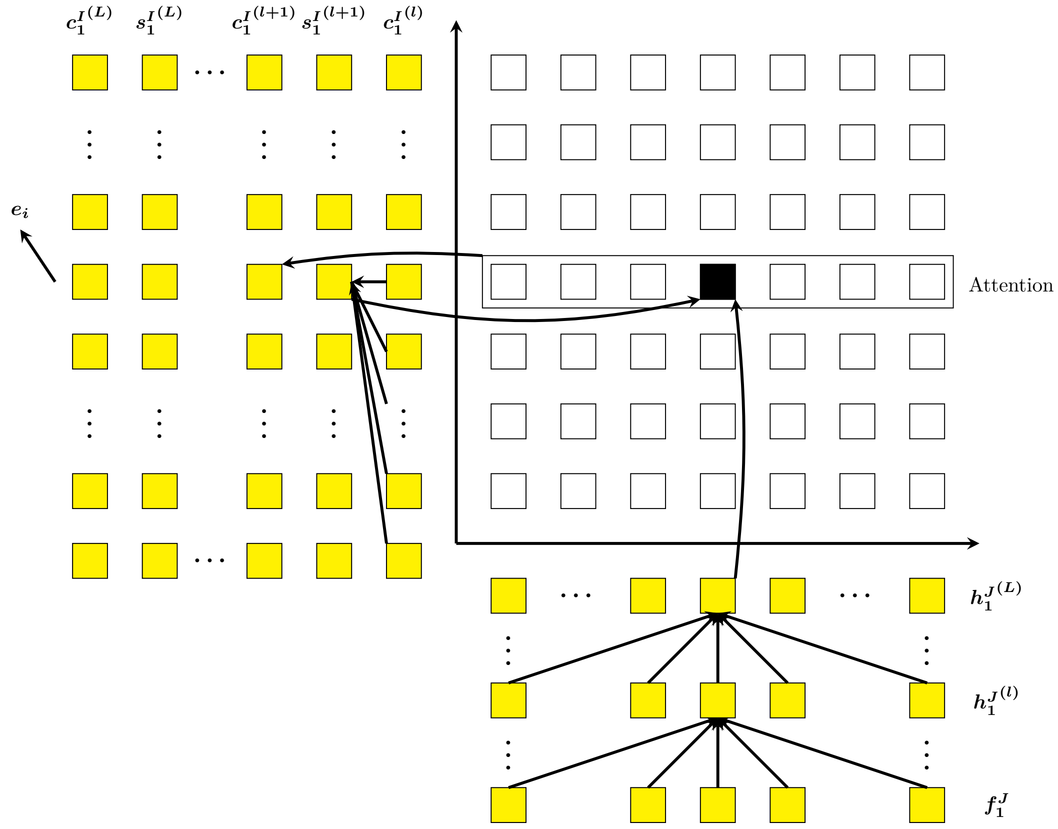
$$p(a_1^S|x_1^T) = \sum_{s_1^T} p\Big(s_1^T, a_1^S | h_1^T(x_1^T)\Big) = \sum_{s_1^T} \prod_t p\Big(s_{t+1}, y_t = a_{s_t} \Big| s_t, a_0^{s_t-1}, h_1^T(x_1^T)\Big)$$

details: RWTH papers at ICASSP and Interspeech

Sequence-to-Sequence Processing: Transformer Approach (Google [Vaswani & Shazeer$^+$ 17])

designed for a 'two-dim.' problem
with input and output sequences:

- keep the *cross-attention* between output and input as in RNN attention [Bahdanau & Cho$^+$ 15]
- for input and output sequence: replace RNN structure by *self-attention*, i. e. pair-wise associations



2020 OpenAI: transformer GPT-3:
– 96 layers, each with 12.288 nodes
– 96 attention heads
in total: 175 Bio parameters

consider MT to be a 1-dim. LM problem:
[input, output] sequences $\rightarrow$ single stream
2013 [Kaltenbrenner & Blunsom 13]
2014 [Sutskever & Vinyals$^+$ 14]
today: GPT successful for many NLP tasks
(generative tasks, beyond MT)

Machine Translation (MT): History

statistical/data-driven approaches were controversial in MT (and other NLP tasks):

- **1969 Chomsky:**
  ... the notion 'probability of a sentence' is an entirely useless one, under any known interpretation of this term.
- **result: mainstream research had a *strict dichotomy* until (around) 2000:**
  – speech = spoken language: signals, subsymbolic, machine learning
  – language = written text: symbols, grammars, rule-based AI
- **until 2000: mainstream approach was rule-based**
  – result: huge human effort required in practice
  – problems: coverage and consistency of rules
- **1989-93: IBM Research: statistical approach to MT**
  1994: key people (R. Mercer, P. Brown) left for a hedge fund
- **1996-2002 RWTH: improvements beyond IBM's approach:**
  – HMM alignments, log-linear modelling, phrases as basic units
  – superior results in DARPA/NIST evaluations
- **around 2004: from singularity to mainstream**
  – F. Och (and more RWTH PhD students) joined Google
  – 2008: service *Google Translate*
- **since 2014: neural MT (unlike count-based MT):**
  attention mechanism [Bahdanau & Cho^{+} 15]

Open Questions

– why do we use Bayes decision rule for ASR ?
– what is the relation of the ANN framework with Bayes decison rule?
– what is the role of softmax output layer in ANNs ?
– what is the relation of training criteria with Bayes decison rule/classification error ?
– what is the relation between training criteria and end-to-end modelling ?
– why should we separate acoustic model and language model ?
– how to use ANNs for acoustic modelling? suitable ANN structures?
– what are synchronization/alignment methods for acoustic modelling ?
– how to use ANNs for language modelling? suitable ANN structures ?

2 Unifying Framework: Bayes Decision Rule

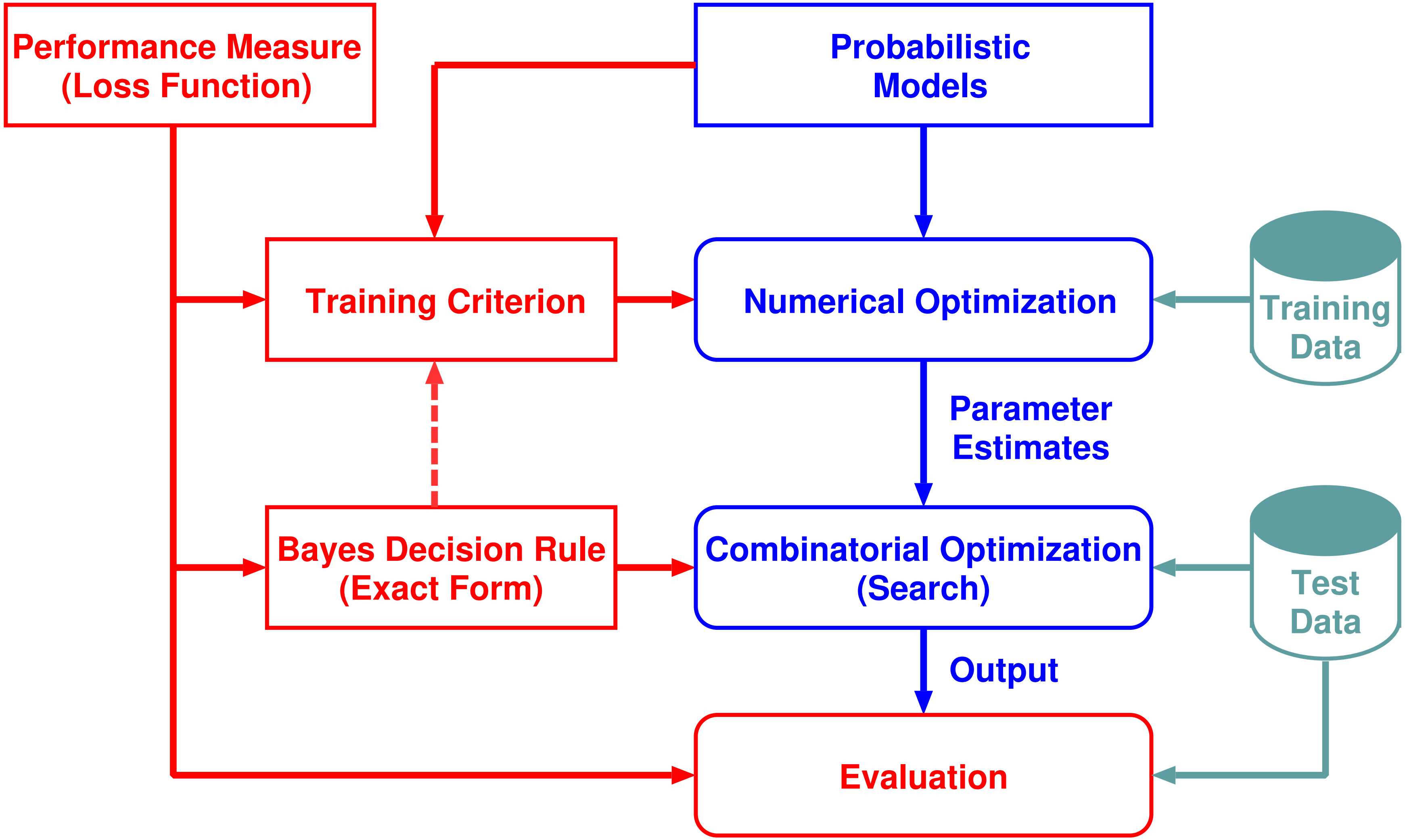
Unifying Framework:
Statistical Decision Theory and Bayes Decision Rule

- so far: historical review of ASR (along with MT) and ANNs
  covering a variety of ANN models and training criteria
- what about training criteria?
  (e. g. cross-entropy, seq.disc. training, state-level min. Bayes risk, expected risk, ...)

  ultimate justification should be based on performance
  – consequence: re-visit Bayes decision rule und its framework
  – example: textbook by Duda & Hart 1973, pp. 11-16
  – originally not explicitly meant for ASR or string processing
- what is not well covered in textbooks or papers:
  – mathematical relation between training criteria and loss function/performance
  – practical implications for training criteria

references, mostly RWTH:
[Ney 03, Schlüter & Scharrenbach$^+$ 05, Xu & Povey$^+$ 10, Schlüter & Nussbaum$^+$ 11],
[Schlüter & Nussbaum$^+$ 12, Schlüter & Nussbaum-Thom$^+$ 13, Schlüter & Beck$^+$ 19]

Unifying View:
Bayes Decision Theory and Machine Learning
(*Why are we doing what we are doing?*)



Principles: Bayes Decision Theory for HLT

- general principles formulated already in 1970s (or before):
  textbook: [Duda & Hart 73, pp. 11-16]
  not explicitly for string processing
- concept: imagine a "huge huge" database of (input,output) string pairs $[x, c]$:
  $$[x_r, c_r], \quad r = 1, ...R$$
- define empirical distribution: $$pr(x, c) = \frac{1}{R} \cdot \sum_{r=1}^{R} \delta(x, x_r)\delta(c, c_r)$$

  remarks:
  – fully specified, no free parameters
  – derived distributions: $pr(c)$, $pr(x)$, $pr(c|x)$, $pr(x|c)$
  – easy principle (i. e. a "huge" table), but difficult implementation for strings
  – simplifying assumption about input x: discrete rather than cont.-valued $x \in \mathbb{R}^D$
- guessing game: knowing x guess c:
  $$x \rightarrow c = \hat{c}(x)$$

  terminology:
  classify the input data (ASR, HTR) or generate the output data (MT)

- perfect solution is not possible:
  - we want to convert a relation $[x, c]$ into a function $x \rightarrow c = \hat{c}(x)$
  - for each pair $[x, c]$, we want to compare: $c \stackrel{?}{=} \hat{c}(x)$
    and thus we need an error measure or loss function $L[c_r, \hat{c}(x_r)], \; r = 1, ..., R$

- popular error measures for strings
  (sequence of symbols: words, subword units, graphemes/letters, phonemes):
  - in general: 0/1 loss function = string error:
    is the string correct as a whole?
  - strings in ASR/HTR: WER = word ("symbol") error rate
    WER = edit distance = errors: ins + del + sub
  - strings in MT: TER = translation error rate
    TER = edit distance + swaps of symbol groups

    alternative: BLEU (more complex)

- key question: how to generate the output string?
  - perfect solution is not possible
  - best compromise: for each input x (which might exist in several pairs $[x = x_r, c_r]$),
    select an output that minimizes the expected loss/risk:

$$x \rightarrow c_*(x) \; := \; \arg\min_c \Big\{ \sum_{\tilde{c}} pr(\tilde{c}|x) \cdot L[\tilde{c}, c] \Big\}$$

    using the class posterior distribution $pr(c|x)$ of the data

Decision Rules: Bayes, Pseudo Bayes and Approximations

- **Bayes decision rule:**

$$x \rightarrow c_*(x) \;:=\; \arg\min_c \Big\{ \sum_{\tilde{c}} pr(\tilde{c}|x) \cdot L[\tilde{c}, c] \Big\}$$

shortcomings in practice:
– difficult/impossible to store $pr(c|x)$
– generalization (from closed to open world): how to handle unseen inputs x ?

- **replace the empirical distribution $pr(c|x)$ by a model $p_\vartheta(c|x)$ ("pseudo Bayes") with parameters ϑ to be learned from data (e. g. neural net):**

$$x \rightarrow c_\vartheta(x) \;:=\; \arg\min_c \Big\{ \sum_{\tilde{c}} p_\vartheta(\tilde{c}|x) \cdot L[\tilde{c}, c] \Big\}$$

- **special choice of loss function: 0/1 = string error:**

$$L[\tilde{c}, c] \;=\; 1 - \delta(\tilde{c}, c) \;\in \{0, 1\}$$
$$x \rightarrow c_\vartheta(x) \;:=\; \arg\max_c \Big\{ p_\vartheta(c|x) \Big\}$$

– terminology: MAP rule (MAP = maximum a-posteriori)
– starting point in most systems
– strictly speaking: adequate only for string error

Bayes Decision Rule and Loss Function

goal: to study the effect of the loss function

three types of outputs and associated loss functions:

- **"atomic" output: 0/1 loss**
  system output has no 'internal structure',
  i. e. single symbols or string as a whole
- **strings with synchronization: Hamming distance**
  loss function: equivalent to symbol error for each position of output string
- **strings with no synchronization: general loss (maybe metric)**
  edit distance (WER) and generalizations (TER)

note minimalistic notation:
– single (class) symbol: c or c_n
– several variables of the same type: $c, \tilde{c}, c', ...$
– string of symbols: c or $c_1^N = c_1...c_n...c_N$
– decision rule generating an output: $x \rightarrow \hat{c}(x)$

Strings with Synchronization: Symbol Error

correct string:	$\tilde{c}_1$	$\tilde{c}_2$	...	$\tilde{c}_{n-1}$	$\tilde{c}_n$	$\tilde{c}_{n+1}$	...	$\tilde{c}_{N-1}$	$\tilde{c}_N$
	|	|	|	|	|	|	|	|	|
hypothesized string:	c_1	c_2	...	c_{n-1}	c_n	c_{n+1}	...	c_{N-1}	c_N

two types of posterior distributions:

joint: $p(c_1^N|x_1^N)$ **marginal:** $p_n(c_n|x_1^N) := \sum_{\tilde{c}_1^N : c_n = \tilde{c}_n} p(\tilde{c}_1^N|x_1^N)$

decision rule for minimum symbol error
using Hamming distance (= symbol error in each position n):

$$L[\tilde{c}_1^N, c_1^N] := \sum_n [1 - \delta(\tilde{c}_n, c_n)]$$

$$x_1^N \rightarrow \hat{c}_1^N(x_1^N) = \arg\min_{c_1^N} \left\{ \sum_{\tilde{c}_1^N} p(\tilde{c}_1^N|x_1^N) \; L[\tilde{c}_1^N, c_1^N] \right\} = \ldots$$

$$= \left[\arg\max_{c_n} \left\{ p_n(c_n|x_1^N) \right\} \right]_{n=1}^N$$

compare with minimum string error:

$$x_1^N \rightarrow \hat{c}_1^N(x_1^N) = \arg\max_{c_1^N} \{ p(c_1^N|x_1^N) \}$$

given synchronization: $\left[c_1^N, x_1^N\right] = \left[c_n, x_n\right]_{n=1}^N$

input vectors:	x_1	x_2	...	x_{n-1}	x_n	x_{n+1}	...	x_{N-1}	x_N
	|	|	|	|	|	|	|	|	|
output symbols:	c_1	c_2	...	c_{n-1}	c_n	c_{n+1}	...	c_{N-1}	c_N

$$p_n(c|x_1^N) = \sum_{c_1^N : c_n = c} p(c_1^N | x_1^N)$$

missing synchronization (ASR) between x_1^T and x_1^N:

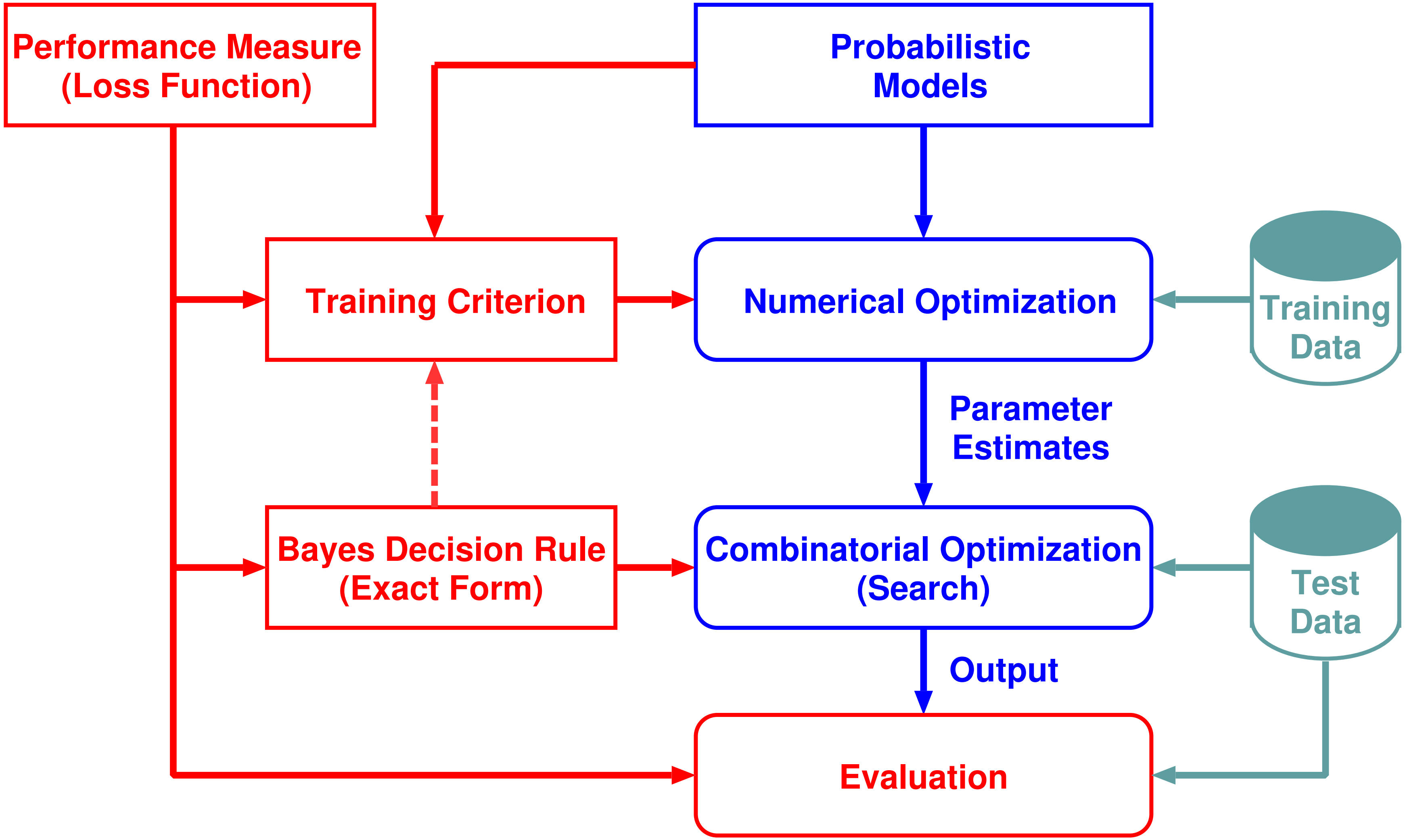
input vectors:	x_1	x_2	...	...	x_{t-1}	x_t	x_{t+1}	...	...	x_{T-1}	x_T
			?		?		?		?		
output symbols:		c_1	c_2	...	c_{n-1}	c_n	c_{n+1}	...	c_{N-1}	c_N	

approximative synchronization using a seed string $\hat{c}_1^N = \hat{c}_1^N(x_1^T)$ (e. g transcription):

$$p_n(c|x_1^T) \;=\; ?? \quad \cong \sum_{c_1^N : c_n = c} p(c_1^N | \hat{c}_1^N, x_1^T)$$

related to training criteria: Povey's minimum word error rate,
state-level minimum Bayes risk (sMBR), expected Bayes risk, ...

Unifying View:
Bayes Decision Theory and Machine Learning
(*Why are we doing what we are doing?*)



mathematical analysis (omitting details):

- ***Bayes decision rule:*** **effect of loss function**
  - **compare 0/1 loss with general loss** $L[\tilde{c}, c]$
  - **identical results for metric loss function** $L[\tilde{c}, c]$ **(e. g. edit distance)**

    if $\max_c p_\vartheta(c|x) \geq 0.5$

    note: purely mathematical result
    [Schlüter & Scharrenbach$^+$ **05, Schlüter & Nussbaum**$^+$ **11]**

- ***training critera*** **for model** $p_\vartheta(c|x)$
  - **should be formulated as a function of model** $p_\vartheta(c|x)$
  - **should interpret the model as an approximation to the true distribution** $pr(c|x)$
  - **should be related to performace (expected loss)**
  - **training in practice: HUGE numerical optimization problem**
    (many shortcuts and approximations beyond cross-entropy training)

mathematical analysis for string error (0/1 loss) [Ney 03]:

- **empirical (= true) distributions $pr(c,x)$ and $pr(c|x)$ (as defined by training data: $c_r, x_r,\ r = 1, ..., R$):**

  $E_* = 1 - A_* =$ **true Bayes classification error: absolute minimum using true Bayes rule:** $x \rightarrow \hat{c}_*(x) = \text{argmax}_c\{pr(c|x)\}$ $\quad A_* = \sum_x pr(x) \cdot pr\big(c = c_*(x)|x\big)$

- **model $p_\vartheta(c|x)$ (e. g. an ANN) with set of parameters ϑ:**

  $E_\vartheta = 1 - A_\vartheta =$ **model-based classification error using: pseudo Bayes rule:** $x \rightarrow \hat{c}_\vartheta(x) = \text{argmax}_c\{p_\vartheta(c|x)\}$ $\quad A_\vartheta = \sum_x pr(x) \cdot pr\big(c = c_\vartheta(x)|x\big)$

upper bound: Kullback-Leibler divergence (relative entropy):

$$1/2 \cdot \big[E_* - E_\vartheta\big]^2 \leq \sum_x pr(x) \sum_c pr(c|x) \log \frac{pr(c|x)}{p_\vartheta(c|x)} = \frac{1}{R} \sum_{r=1}^{R} \log \frac{pr(c_r|x_r)}{p_\vartheta(c_r|x_r)}$$

criterion: minimize this upper bound over ϑ: $\rightarrow$ cross-entropy criterion
(other upper bounds: binary divergence and squared error)

more realistic situation:
– word/symbol errors (edit distance) in lieu of string errors
– no closed-form solution: approximations required

ASR Modelling: String Posterior Probability

- complete model for [input,output] pair $[x_1^T, W = a_1^S]$
  consists of language model (LM) and acoustic (-phonetic) model (AM):

$$p_\vartheta(W|x_1^T) := \frac{q_\vartheta^\alpha(W) \cdot q_\vartheta^\beta(W = a_1^S|x_1^T)}{\sum_{\tilde{W}} q_\vartheta^\alpha(\tilde{W}) \cdot q_\vartheta^\beta(\tilde{W} = \tilde{a}_1^S|x_1^T)}$$

  with model parameters ϑ (and exponents α, β)

- motivation: the log-linear combination mimicks the generative approch:

$$p_\vartheta(W|x_1^T) := \frac{p_\vartheta(x_1^T, W)}{\sum_{\tilde{W}} p_\vartheta(x_1^T, \tilde{W})} = \frac{p_\vartheta(W) \cdot p_\vartheta(x_1^T|W)}{\sum_{\tilde{W}} p_\vartheta(\tilde{W}) \cdot p_\vartheta(x_1^T|\tilde{W})} = \frac{p_\vartheta(W) \cdot pp_\vartheta(W|x_1^T)}{\sum_{\tilde{W}} p_\vartheta(\tilde{W}) \cdot pp_\vartheta(\tilde{W}|x_1^T)}$$

  with a re-normalized *pseudo posterior*: $pp_\vartheta(W|x_1^T) := 1/Z(x_1^T) \cdot p_\vartheta(x_1^T|W)$

- language model:
  learned from text data only (without annotation) (e. g. 1000 Mio words)

- acoustic model (HMM, finite-state transducer, cross-attention model,...):
  learned from (manually) transcribed audio data (e. g. 1000 hours = 10 Mio words)

Acoustic Model: Training Criterion and Procedure

suitable training criterion for *string errors* with (audio, text) pairs $[X_r, W_r],\ r = 1, ..., R$:

$$\max_{\vartheta} \Big\{ \sum_r \log\, p_\vartheta(W_r|X_r) \Big\} \qquad\qquad p_\vartheta(W|X) \;=\; \frac{q^\alpha(W) \cdot q_\vartheta^\beta(W|X)}{\sum_{\tilde{W}} q^\alpha(\tilde{W}) \cdot q_\vartheta^\beta(\tilde{W}|X)}$$

numerical optimization problem in training:

- ***string errors:* ignore denominator: simplified baseline**
  - **effect: decoupling of AM and LM**
    advantage: independent training of AM and LM
  - **variants for AM training: full sum or best path/Viterbi (frame-wise CE)**
    note: EM framework still works for neural HMM !

- **keep denominator: *sequence discriminative training***
  result: LM affects training of AM !
  - **loss function: *string errors* (IBM 1986: MMI)**
  - **loss function: *symbol* errors (e.g. WER) in string context**
    variants in ASR: Povey's phoneme/symbol error, sMBR, expected loss, ...

  denominator: how to approximate it?
  - **word hypothesis lattice**
  - **simplifed language model (lattice-free MMI, Povey 2016)**

history: Bahl/IBM 1986, Normandin 1991, Valtchev 1996, Povey 2002/16, Heigold 2005/12

ASR: *End-to-End* Approaches

reconsider training criterion for (audio,text) pairs $[X_r, W_r],\ r = 1, ..., R$:

$$\max_{\vartheta} \Big\{ \sum_r \log\ p_\vartheta(W_r|X_r) \Big\} \qquad\qquad p_\vartheta(W|X) := \frac{q^\alpha(W) \cdot q_\vartheta^\beta(W|X)}{\sum_{\tilde{W}} q^\alpha(\tilde{W}) \cdot q_\vartheta^\beta(\tilde{W}|X)}$$

terminology: What does *end-to-end* mean?

- **training criterion: a single global criterion for optimum performance, independent of model structure**
- **monolithic structure of a model:
simplicity/elegance of programming? what about adequacy/performance?**

remarks:

- **ASR: training of acoustic model and language model:**
  – transcribed audio: 1000 hours = 10 Mio words
  – text (from press, books, internet,...): 1000 Mio words and more
- ***end-to-end* concept:**
  **– for training and search/generation: yes
  (? and robustness/easiness of training)**
  – for the structure: can it reflect the training data situation?
  – in addition to LM: pronunciation lexicon?

Effect of AM, Training Criterion and LM (Tüske et al. RWTH 2017)

RWTHAACHEN UNIVERSITY

QUAERO task, English Eval 2013:
broadcast news/conversations, podcasts, TED lectures

Word error rates [%] on QUAERO English Eval 2013
(PP: perplexity of LM = power of LM $\cong$ effective vocab.size)

Acoustic Model (AM): hybrid HMM			Language Model (LM)	
Type		Training Criterion	Count	Count + ANN
			PP=131.1	PP=92.0
Gaussian mixtures		max.lik.	20.7	
		seq.disc. training	19.2	16.1
Neural Net	FF MLP	frame-wise CE	11.6	
		seq.disc. training	10.7	9.0
	LSTM RNN	frame-wise CE	10.6	
		seq.disc. training	9.8	8.2

observations:
– improvements by acoustic ANNs: 50% relative
– improvement by language model ANN: 15% relative
– total improvements by deep learning: 60% relative (from 19.2% to 8.2%)

ASR: Librispeech Task: Hybrid HMM vs. Attention (RWTH 2019)

RWTH AACHEN UNIVERSITY

speech data: read audiobooks from the LibriVox project
with training data:
– acoustic model: 960 hrs of speech
– language model: 800 million words

word error rates [%]:

		WER (dev)		WER (test)	
team	approach	1st half	2nd half	1st half	2nd half
Irie, Zeyer et al. RWTH (Interspeech 2019)	attention with BPE units, 'no' LM	4.3	12.9	4.4	13.5
	+ LSTM-RNN LM	3.0	9.1	3.5	10.0
	+ transformer LM	2.9	8.8	3.1	9.8
Lüscher, Beck et al. RWTH (Interspeech 2019)	hybrid HMM, CART, 4g LM	4.3	10.0	4.8	10.7
	+ seq. disc. training	3.7	8.7	4.2	9.3
	+ LSTM-RNN LM	2.4	5.8	2.8	6.2
	+ transformer LM	2.3	5.2	2.7	5.7

Zeghidour et al., FB 2018	gated CNN with letters/words	3.2	10.1	3.4	11.2
Irie et al., Google 2019	attention with WPM units	3.3	10.3	3.6	10.3
Park et al., Google 2019	attention ... data augmentation	-	-	2.5	5.8

Acoustic Modelling: Recent Results on Librispeech Task (RWTH 2022 - 2023)

word error rates [%]: recent results by RWTH team
(W. Zhou, S. Berger, T. Raissi, M. Zeineldeen, ...)
– acoustic encoder: conformer
– language model: transformer

			WER [%] (test)	
method	parameters	epochs	clean	other
hybrid HMM (phonemes, CART)	86M	11	2.2	4.5
transducer with phonemes (context 1)	75M	36	1.9	4.0
transducer with BPE units (context 1)	87M	56	1.8	4.1
transformer with BPE units (full context)	103M	100	1.9	4.2

word error rates [%] of other teams:

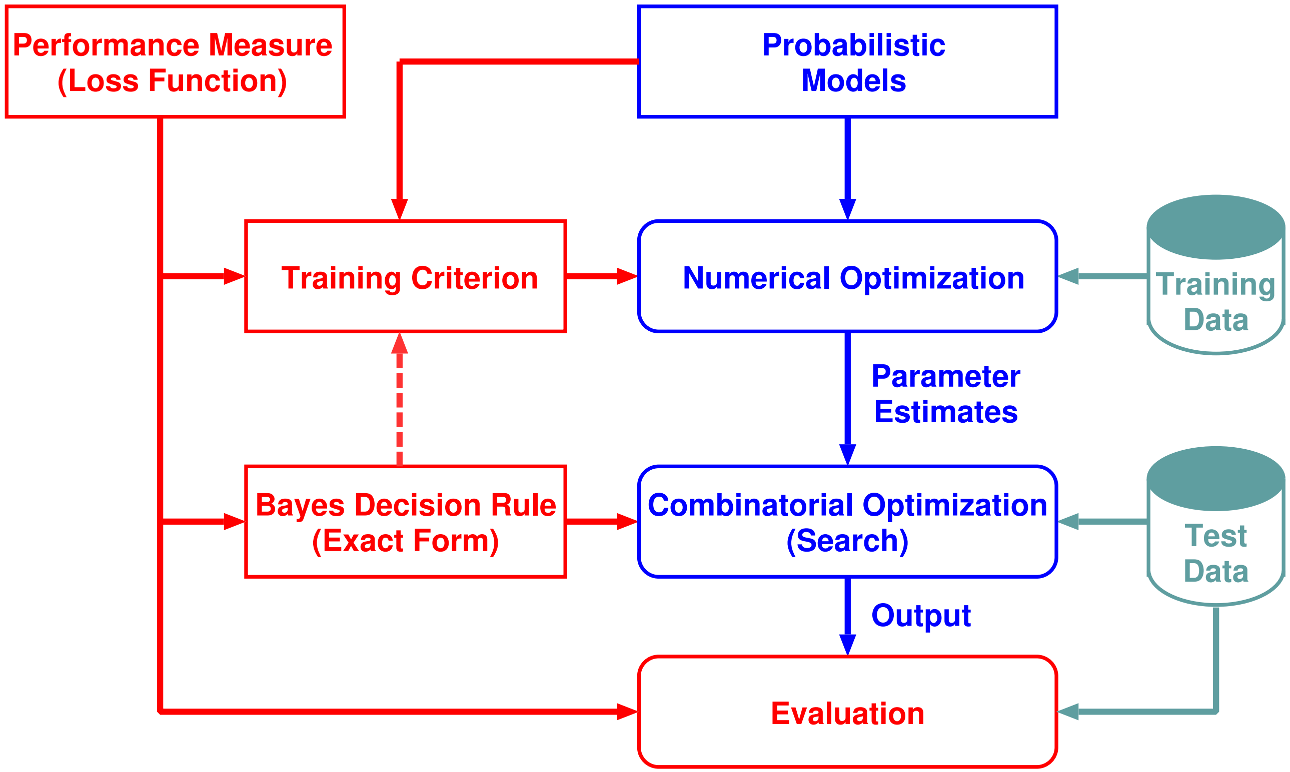
			WER [%] (test)	
authors/method	parameters	epochs	clean	other
Park & Zhang^{+} 2020: transformer	360M	600	2.2	5.2
Zhang & Wang^{+} 2020: CTC-transformer	124M	200	2.1	4.2
Kim & Wu^{+} 2023: transformer	149M	80	1.8	3.7

Statistical Decision Theory for ASR (and NLP): Where do we stand now ?

RWTH AACHEN UNIVERSITY

- exact loss function:
  - not so important in testing
  - more important in training
- probabilistic models:
  - are most important: caused progress 1980-2023
  - dependencies and synchronization between input/output strings
  - often (e. g. ASR): separate LM
- training criterion:
  - is important
  - depends on prob. models
- numerical optimization:
  - hard math. problem
  - all variants of backpropagation
  - important in practice (1992 vs. 2022!)
- decision rule: search/generation: today's models: more important for low-accuracy conditions

this lecture:
– statistical decision theory defines a perfect framework
– its principles go beyond NLP and ANN



AppTek

3 Language Models (small and large)

Language Modelling in ASR

Bayes decision rule for generating word sequence w_1^N from speech signal x_1^T (assuming a log-linear model and dropping the denominator):

$$x_1^T \to \hat{w}_1^{\hat{N}}(x_1^T) \;=\; \underset{N, w_1^N}{\operatorname{argmax}} \left\{ q_\vartheta^\alpha(w_1^N) \cdot q^\beta(w_1^N | x_1^T) \right\}$$

language model: the prior probability $q_\vartheta(w_1^N)$ and its parameters ϑ

observations about the language model $q_\vartheta(w_1^N)$:
– it can be learned from text only (unlabeled data!), e. g. from 100 Mio to 10 Bio words
– it can improve performance dramatically

question:

How to measure the quality of an LM (without a recognition experiment)?

Quality of Language Model and Perplexity

considerations:

- **use prior $q_\vartheta(w_1^N)$ in Bayes decision rule, but it depends on the single sentence and its length**
- **define a sufficiently large test corpus by concatenating all test sentences to a LONG super sentence (use special symbols for sentence end and unknown word)**
- **apply the LM probability to this super sentence of N words and perform normalization:**
  – geometric average of probability per word by computing N-th root
  – invert average probability into perplexity: = average *effective* vocabulary size

formal definition of perplexity PP:

$$PP := \Big(q_\vartheta(w_1^N)\Big)^{-1/N} = \Big(\prod_{n=1}^N q_\vartheta(w_n|w_0^{n-1})\Big)^{-1/N}$$

$$\log PP = -\frac{1}{N} \cdot \sum_{n=1}^N \log q_\vartheta(w_n|w_0^{n-1})$$

with artificial start symbol w_0

interpretation of perplexity: from single sentence to whole database

prior probability $q_\vartheta(w_1^N)$ of any sentence $w_1^N = w_1...w_n...w_N$
based on simplified dependence: word trigram language model:

$$q_\vartheta(w_1^N) \;=\; \prod_{n=1}^{N} q_\vartheta(w_n|w_1^{n-1}) = \prod_{n=1}^{N} q_\vartheta(w_n|w_{n-2}, w_{n-1})$$

disambiguation of homophones (Tangora system, IBM 1985):

- homophones: two, too, to

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

- homophones: write, Wright, right

  Please write to Mrs. Wright right away.

Language Modelling: Approaches

- **limited history: Markov chain of order k:**
  limit the dependence on the full history w_0^{n-1} to the immediate k predecessor words:

$$q_\vartheta(w_n|w_0^{n-1}) \;:=\; q_\vartheta(w_n|w_{n-k}^{n-1})$$

  modelling concepts:
  – discrete: event counts (e. g. word fourgrams, trigrams, bigrams, unigrams) and smoothing
  – continuous-valued: FF-MLP with word embeddings (IMPORTANT!), i. e. a mapping from word symbols to vectors

- **unlimited history (with word embeddings):**
  continous-valued: RNN and other sequence models (e. g. transformer)

natural training criterion for a corpus w_1^N: minimum perplexity

$$\max_\vartheta \left\{ \sum_{n=1}^{N} \log \; q_\vartheta(w_n|w_0^{n-1}) \right\}$$

– equivalent to cross-entropy training (or *perplexity*, maximum likelihood)
– resulting estimates: relative frequencies based on event counts

Neural Language Modelling
[Sundermeyer et al.; RWTH 2012, 2015]

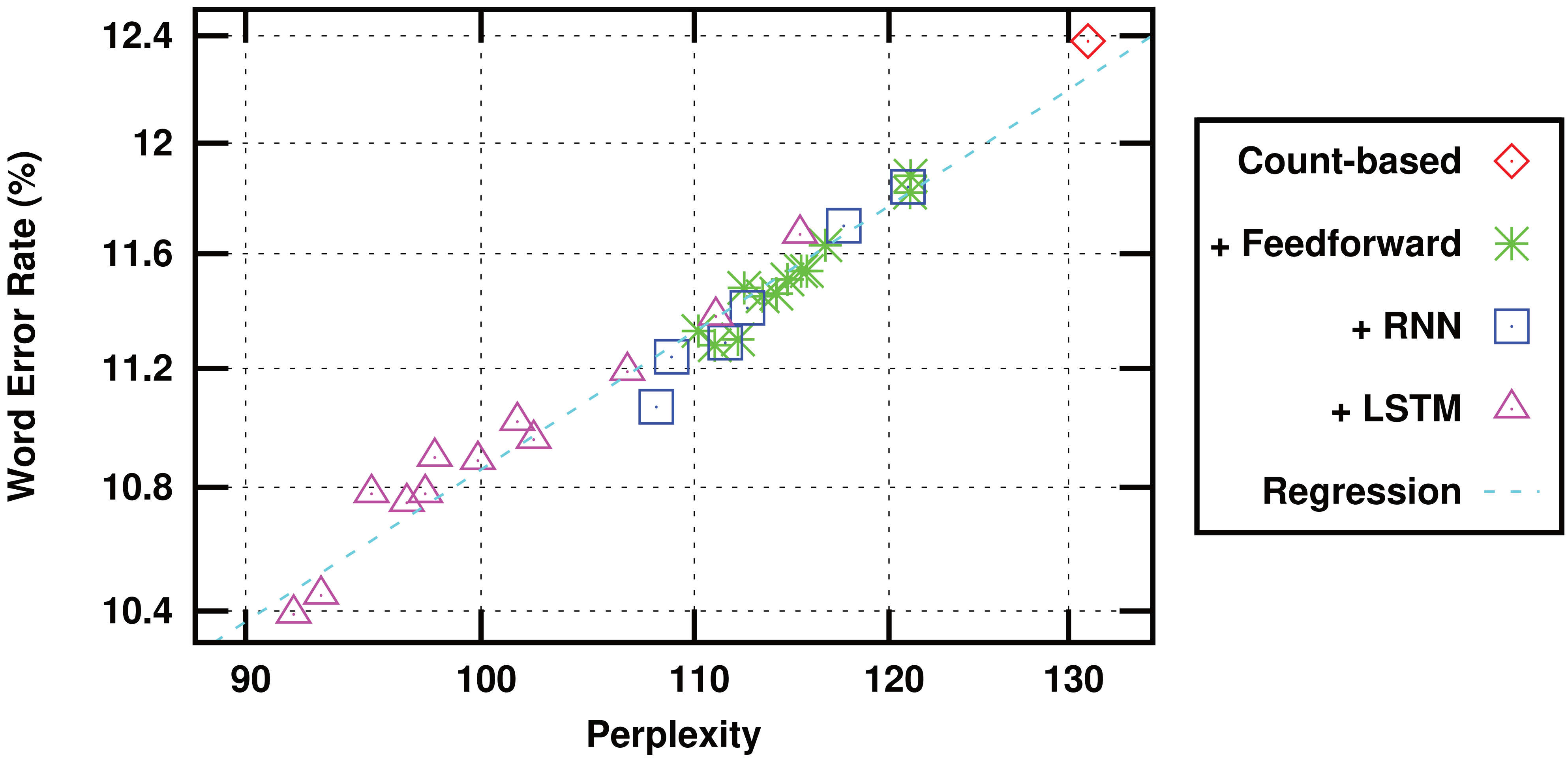
- important principle (undervalued!):
  - move away from count-based statistics for categorial random variables
  - instead: word/symbol embeddings and operations in a high-dim. vector space
- interpolation of TWO models (2015):
  count model (3 Bio words) + ANN model (60 Mio words)
- details and refinements:
  - use of word classes for softmax in output layer
  - unlimited history of RNN: requires re-design of ASR search
- perplexity (PP) and word error (WER) rate on test data (QUAERO)

models	PP	WER[%]
count model	131.2	12.4
+ 10-gram MLP	112.5	11.5
+ Recurrent NN	108.1	11.1
+ LSTM-RNN	96.7	10.8
+ 10-gram MLP with 2 layers	110.2	11.3
+ LSTM-RNN with 2 layers	92.0	10.4

- improvements achieved:
  - perplexity: 30% reduction: from 131 to 92
  - WER: 15% reduction: from 12.4% to 10.4%

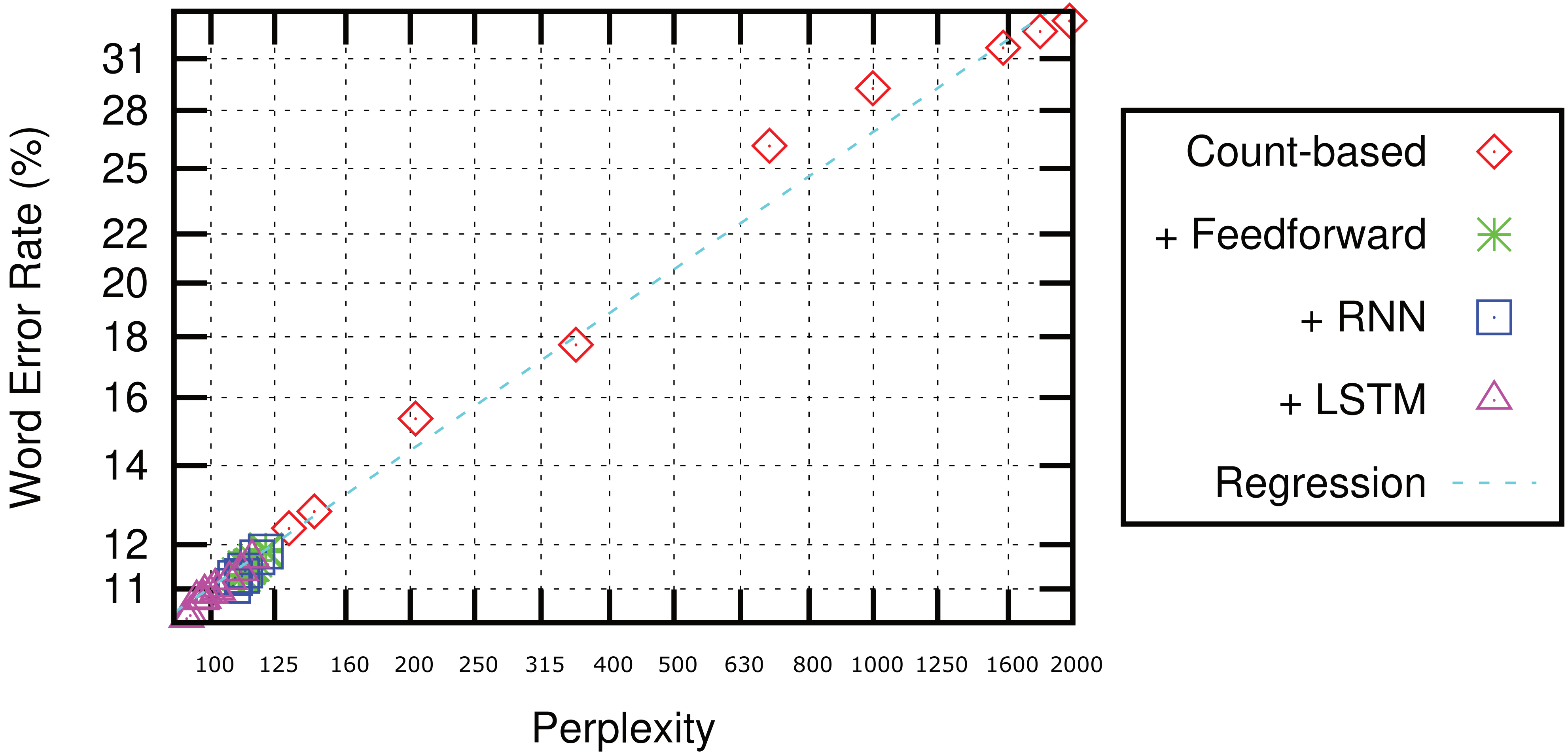
Effect of Language Model: Word Error Rate vs. Perplexity

empirical law: $WER = \alpha \cdot PP^{\beta}$ **with** $\beta \in [0.3, 0.5]$
[Makhoul & Schwartz 94, Klakow & Peters 02]



Effect of Language Model: Word Error Rate vs. Perplexity

empirical law: $WER = \alpha \cdot PP^{\beta}$
open question: theoretical justification?



note: Google paper at ICASSP-23: LLM for ASR

encryption method: homophonic ciphers:
each plaintext letter is mapped to one or several ciphertext symbols.
compare with spoken language:
a *homophone* (= pronunciation) has several different writings.

encrypted texts: two examples:
Beale ciphers (Virginia/US 1820/85) and Zodiac killer ciphers (Bay Area/US 1968/9)
– Beale cipher 2: sequence of 762 numbers with 182 distinct numbers
– Zodiac killer 408-cipher: sequence of 408 'artificial' symbols with distinct 54 symbols

(sort of) perfect decipherment:

- letter-based language model (of general English) is used to score all possible substitution possibilities
- combinatorial search problem: beam search
- paper at EMNLP 2014: M. Nuhn, J. Schamper, H. Ney: *Improved Decipherment of Homophonic Ciphers.*
- article in *Mental Floss*, 04-Jun-2018:
  `https://www.mentalfloss.com/article/540277/beale-ciphers-buried-treasure`

History:

- **1989 [Nakamura & Shikano 89]:**
  English word category prediction based on neural networks.
- **1993 [Castano & Vidal$^+$ 93]:**
  Inference of stochastic regular languages through simple recurrent networks
- **2000 [Bengio & Ducharme$^+$ 00]:**
  A neural probabilistic language model
- **2002 [Schwenk & Gauvain 02, Schwenk 07]: Continuous space language models**
- **2010 [Mikolov & Karafiat$^+$ 10]:**
  Recurrent neural network based language model
- **2012 RWTH Aachen [Sundermeyer & Schlüter$^+$ 12]:**
  LSTM recurrent neural networks for language modeling
- **2017 [Vaswani & Shazeer$^+$ 17]: transformer architecture (originally for MT)**
- **since 2019 beyond ASR: multi-lingual, multi-task, many parameters (200 billion!)**
  (GPT, Whisper, LaMDA, OPT, Bloom, ChatGPT, ...):
  – GPT: general pretrained transformer
  – LLM: large-scale language models

important component in ANN-based LMs (contrast: count-based LM):
– word/symbol representations/embeddings: vectors in high-dim. space
– in addition to ANN structures (MLP, RNN, LSTM-RNN, transformer, ...)

word representations used without ANN context
(personal communication, Eduardo Lleida, 13-Nov-2023):

- **1971 Salton: information retrieval using term-document matrix**
- **1993 Schütze & Peterson: co-occurrence of two words**
- **2004 Bellegarda: Latent Semantic Modelling for Speech Recognition**
- **2013 Hofmann: Probabilistic Latent Semantic Analysis**

power of LMs and word representations (spirit of *distributional semantics*):
1954 Harris: Words are similar if they appear in similar contexts.
1957 Firth: You shall know a word by the company it keeps.

- **papers by [Collobert & Weston 08, Collobert & Weston$^+$ 11]:**
  2008: *A Unified Architecture for NLP: Deep Neural Networks with Multitask Learning.*
  2011: *NLP (almost) from Scratch.*

  use of word vectors for *formal* NLP tasks:
  POS/NER tagging, syntactic analysis, semantic role labeling, text classif., ...

- **word vectors: (semantic) interpretations and calculations**
  examples of relations between word vectors [Mikolov & Corrado$^+$ 13]:

$$Germany - Berlin \cong France - Paris$$
$$king - queen \cong man - woman$$

- **2013/2014: use LM concept for MT [Kaltenbrenner & Blunsom 13, Sutskever & Vinyals$^+$ 14]**

- **since 2019: LLMs (large-scale LMs) based on GPT architecture:**
  – G: generative: generate text (as opposed to formal NLP tasks)
  – P: pre-trained: based on text without *any* annotation
  – T: transformer: ANN structure for sequence-to-sequence processing

  LLM implies: more data, more parameters (200 Bio), multi-lingual, multi-task, ...

Refining LLMs: *InstructGPT*

***InstructGPT* introduced by OpenAI, arxiv, 04-Mar-2022:**
Training language models to follow instructions with human feedback.

three levels of training:

- **pre-training or unsupervised training (using log perplexity):**
  - **training mode: raw text with no annotation**
  - **operation mode (surprising result !):**
    type of task (*prompt*): can be specified in plain language
    e. g. Q&A, summarization, story generation, *dialog!*, ...
    e. g. *multilingual* LLM: translation
    full system operation is described by a triplet (in plain language!):
    triplet $:=$ [prompt, input, output]
    (typically used in so-called *few-shot learning/conditioning*)
- **supervised fine-tuning:**
  - **training data: based on (many) triplets of the above type**
  - **training criterion: (log) perplexity**
    all triplets are interpreted as a *single* sequence of text
- **human feedback and reinforcement learning:**
  - **starting point: system is used to generate the outputs for [prompt, input] pairs**
  - **human evaluation and ranking for LLM-generated outputs**
  - **reinforcement learning based on human scores**

LLM and GPT: Typical Tasks

every-day NLP tasks with plain text for input and output:

- **text summarization:**
  input: ***full text***
  output: ***text summary***
- **story generation:**
  input: ***key words***
  output: ***full text***
- **machine translation (with bilingual training data):**
  input: ***sentence in source language***
  output: ***sentence in target language***
- **conversational dialog (with many turns):**
  input: ***customer query/command***
  output: ***system response***

remarkable property (in contrast to formal NLP tasks):
everything is expressed in terms of plain every-day language:
– system input: formulated by the user
– type of task (*prompt/instruction*): specified by the user
– generated output: smooth fluent language
(primary goal which a language model is designed for)

ChatGPT and Related Models

- **large-scale language model (LLM) called *chatGPT*:**
  - API introduced on 30-Nov-2022 by OpenAI
  - function: human-like conversational (text) dialog *(unlimited domain)*
  - CEO S. Altman: "costs are eye-watering"
  - operational loss in 2022: 540 Mio USD (416 on computing, 89 on staff)

- **OpenAI's technology behind *chatGPT*:**
  - baseline architecture *GPT: generative pre-trained transformer*
  - *GPT-3*: with 1.3 to 175 Bio parameters,
    trained on 300 Bio (subword) tokens (cut-off date: June 2020)
  - *InstructGPT* (sibling to *ChatGPT*): refinement with human feedback

- **other types of dialog systems:**
  - limited-domain, task-oriented dialog
  - explicit dialog strategy: manually designed and coded

  specific systems: *voice command and control*
  - Amazon's Alexa (loss in 2022: 10 Bio USD - 12 000 employees)
  - Apple's Siri
  - Google's (Digital) Assistant

Some LLMs (until 2022)

- **OpenAI:**
  - 2018 GPT-1: 0,12 Bio
  - 2019 GPT-2: 1,5 Bio
  - 2020 GPT-3: 175 Bio (train: 300 Bio)
  - 2022 *InstructGPT* and *ChatGPT*
- **Google:**
  - 2018 BERT: 3,3 Bio (train: 300 Bio, 40 epochs)
  - 2019 T5: 11 Bio (train: 1000 Bio)
  - 2020 Meena (for dialog): 2,6 Bio (train: 61 Bio)
  - 2022 LaMDA: 137 Bio (train: 2810 Bio)
  - 2022 PaLM: 540 Bio (train: 780 Bio)
- **more LLMs:**
  - 2019 BART / Meta: 0,33 Bio (train: 55 Bio, 40 epochs)
  - 2019 Megatron / Nvidia: 3,9 Bio (train: 366 Bio)
  - 2020 DialoGPT / Microsoft: 0,76 Bio (train: 10 Bio)
  - 2022 OPT / Meta: 175 Bio (train: 180 Bio)
- **years 2021-2022: more than 50 LLMs**
  recent European activities:
  - BLOOM / BigScience: 176 Bio (train: 366 Bio)
  - Luminous / Aleph Alpha (OpenGPT-X): 70 Bio (train: 588 Bio)
  - HPLT (EU project): major EU languages

4 Conclusions

40 years of building operational systems for HLT:

- **success of data-driven vs. handcrafted rule-based approaches**
- **misconception: things started 40 years ago, not in 2013!**
- **persistent evolution of data-driven concepts:**
  - **signal-processing NLP: ASR and HWR**
  - **text-processing NLP:**
    - ☐ **language models for ASR (+ HWR + MT)**
    - ☐ **machine translation (MT)**
    - ☐ **large language models for NLU, e. g. Q&A, dialog management, ...**
- **statistical decision theory:**
  unifying framework for data-driven approach and machine learning:
  - **distinguish ingredients:**
    loss function, prob.model, training criterion along with numerical optimization
  - **includes as a special case: ANNs and deep learning**
  - **most useful framework after 40 years of NLP**

- large-scale language models:
  - primary design goal: to generate smooth fluent text
  - approach: data, but no manual design or coding
  - dialog management: learned by data-driven approach (unlike manually designed dialog strategies)
  - (hopeful) by-product: semantic correctness ?
- LLMs are part of data-driven machine learning:
  - more data, more complex models, more computation
  - 1989 R. Mercer/IBM: *There is no data like more data.*
- specific success ('revolution'):
  - symbol embeddings/vectors in contrast to symbol count statistics along with operations in high-dim. vector space:
  - useful for areas beyond NLP? general concept for categorical statistics?

Success of LLMs: Why?

where does the success/hype of LLMs come from?

- power of transformer architecture
  (and computer hardware!)
- huge amount of training data:
  – no annotation required!
  – straightforward training criterion: perplexity
- instruction/prompt along with input and output:
  everything in every-day language (unlike a formal NLP task)
- in particular: success for dialog tasks:
  no explicit dialog strategy!

- unclear:
  relevance of supervised fine-tuning and reinforcement learning

future: what time horizon: 3, 5, 10, 20 years?
e. g. difficult prediction: ANN around 1990

short-term horizon: low-hanging fruits
more data, more complex models, more parameters, more computation

long-term horizon: scientific challenges:
beyond more data, we need better mathematical frameworks:

- back-propagation search:
  beyond trial and error: better theory of numerical optimization
- present ANN structures
  - deep MLP, RNN, LSTM, self-embedding, transducer, transformer,...:
  - lack of principal mathematical justification:
    why are some structures better for modelling and learning?
- beyond ANN structures:
  - what about going beyond the present structures (matrix-vector product + nonlinearity)?
  - there is plenty of (data-driven) life outside and beyond deep learning!
    (but yes, it will be complex mathematical models)

What about the Future? (ctd)

- **word/symbol embeddings in symbolic processing (NLP):**
  – most important concept in lieu of count-based statistics
  – widely underrated in statistics of categorical data (and general NLP ?)

- **open research directions: beyond *supervised* machine learning:**
  strictly *unsupervised* machine learning,
  i. e. absolutely no parallel (input,output) pairs

END

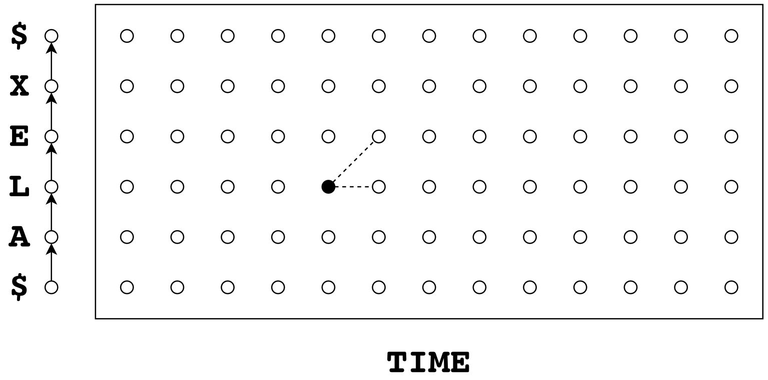
RTTH, Jaca 2023: Data-Driven Speech & Language Technology (HLT): From Small to Large Models

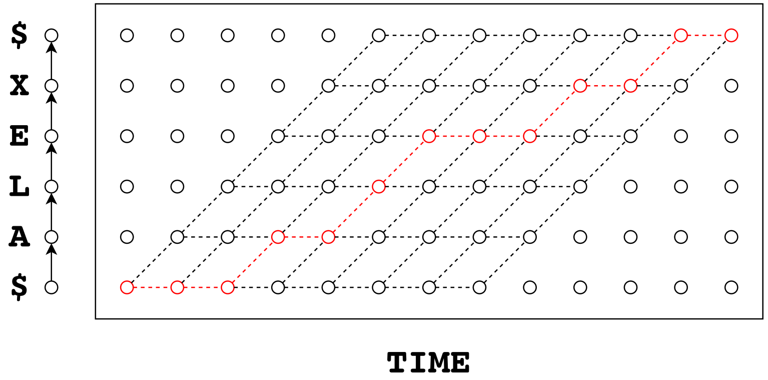
5 Backup Slides

Mathematical Formalism (Alternative): Direct or Posterior HMM for $p(a_1^S|x_1^T)$ (view: how to leave $[t,\ s=s_t]$?)

three sequences over time:

$$x_1^T = x_1, ..., x_t, ..., x_T$$
$$s_1^T = s_1, ..., s_t, ..., s_T$$
$$y_1^T = y_1, ..., y_t, ..., y_T$$





path consists of transitions leaving $[t,\ s=s_t]$: first label y_t and then transition δ_t:

$$[t, s=s_t] \rightarrow [t+1,\ s_{t+1}=s_t+\delta_t] \qquad \delta_t \in \{0,1\}$$

JOINT event of frame label y_t and δ_t:

$$[y_t,\ \delta_t]: \quad p\Big([y_t, \delta_t]\Big|..., x_1^T\Big)$$

link to state s with label $a_s \in a_1^S$:

$$[y_t,\ \delta_t]: \quad p\Big([y_t=a_s, \delta_t]\Big|..., x_1^T\Big)$$

first-order dependence in a_1^S:

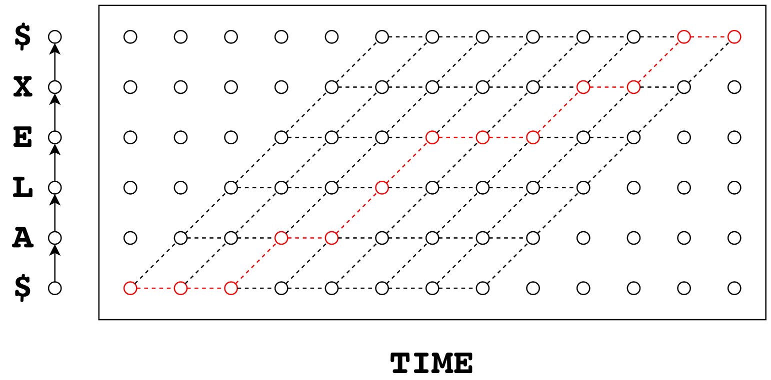
$$[y_t, \delta_t]: \quad p\Big([y_t=a_s, \delta_t]\Big|a_{s-1}, x_1^T\Big)$$

remarks:
- **for full context, replace a_{s-1} by a_0^{s-1}**
- **alternative notation: how to reach $[t,\ s=s_t]$? first transition δ_t and then label y_t:**

$$p\Big([\delta_t, y_t=a_s]\Big|a_{s-1}, x_1^T\Big)$$

Mathematical Formalism (Alternative): Direct or Posterior HMM for $p(a_1^S|x_1^T)$ (view: how to leave $[t,\ s = s_t]$?)

formal derivation of full model:



$$p(a_1^S|x_1^T) \;=\; \sum_{s_1^T} p(a_1^S, s_1^T|x_1^T)$$

finite-state model: factorization over t:

first-order model in s_1^T and a_1^S

$$= \sum_{s_1^T} \prod_t p\big([y_t = a_{s_t}, s_{t+1}] \big| s_t, a_{s_t-1}, x_1^T\big)$$

difference in state/segment indices: $\delta_t := s_{t+1} - s_t$

$$= \sum_{s_1^T} \prod_t p\big([y_t = a_{s_t}, \delta_t] \big| a_{s_t-1}, x_1^T\big)$$

explicit segmental interpretation:

frames t within segment s:
- **last frame: $\delta_t = 1$**
- **other frames: $\delta_t = 0$**

$$= \sum_{s_1^T} \prod_s \prod_{t:\, s_t = s} p\big([y_t = a_s, \delta_t] \big| a_{s-1}, x_1^T\big)$$

acoustic encoder : $h_t = h_t(x_1^T)$

$$= \sum_{s_1^T} \prod_s \prod_{t:\, s_t = s} p\big([y_t = a_s, \delta_t] \big| a_{s-1}, h_t(x_1^T)\big)$$

goal of language modelling: compute the prior $q_\vartheta(w_1^N)$ of a word sequence w_1^N
– how plausible is this word sequence w_1^N (independently of observation x_1^T!) ?
– measure of language model quality: perplexity PP (= geometric average)
interpretation: effective vocabulary size as seen by ASR decoder/search

$$\log PP \;:=\; \log\; 1\Big/\sqrt[N]{q_\vartheta(w_1^N)} \;=\; -1/N \cdot \sum_{n=1}^{N} \log\; q_\vartheta(w_n|w_0^{n-1})$$

interpretation: prediction task:
based on history w_0^{n-1}, predict $q_\vartheta(w_n|...)$

approaches:
– use full history: RNN or LSTM
– truncate history: $\rightarrow$ k-gram MLP

perplexity PP on test data (QUAERO)
(Sundermeyer et al.; RWTH 2012, 2015):

approach	PP
baseline: count model	163.7
10-gram MLP	136.5
RNN	125.2
LSTM-RNN	107.8
10-gram MLP with 2 layers	130.9
LSTM-RNN with 2 layers	100.5

important result: improvement of PP by 40%

most important contributions (see page 5):

- **academia:**
  - general HMM framework
  - RNN-HMM [Robinson 1994]
  - RNN-CTC [Graves 2009]
  - deep learning (in the narrow sense!) [Hinton 2011]
  - cross-attention [Montreal team 2014]
- **industry:**
  - self-attention and transformer
  - conformer

History: How (Small) Language Models Started (1980-2000)

(small) language models:

- **introduced by IBM for ASR around 1980**
  - **key advantage: use of text data without annotation**
  - **statistics: based on counts of word trigrams (and higher order n-grams)**
  - **concept: sucessfully transferred from ASR to HWR and MT**

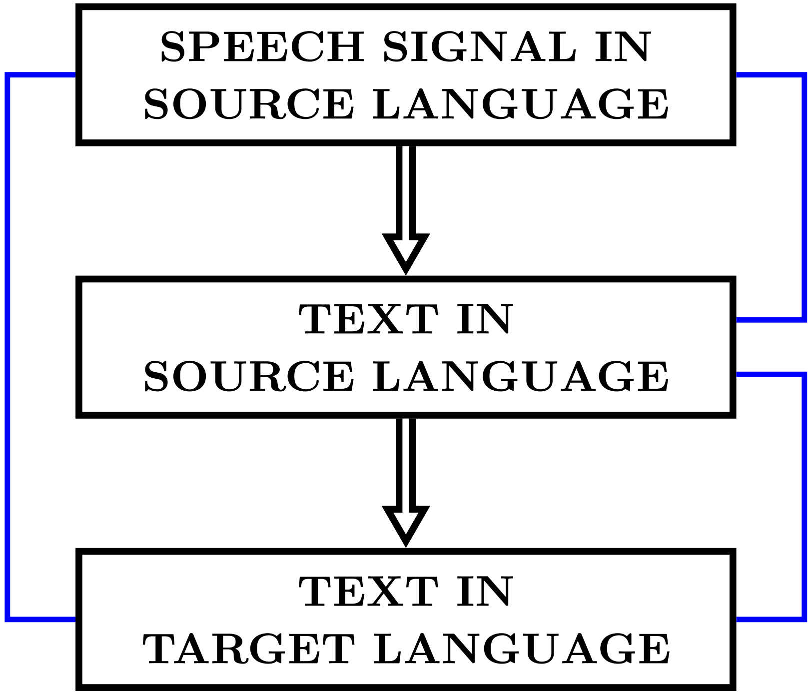
- **experimental conditions around 2000:**
  - **training: about 100 Mio running words (tokens)**
  - **model size: same order of magnitude**

- **training criterion: log perplexity (= cross-entropy), i. e. *predict next word* probability of a word sequence $w_1^N = w_1 ... w_n .... w_N$:**

$$\log\ p_\vartheta(w_1^N) = \sum_{n=1}^{N}\ \log\ p_\vartheta(w_n|w_0^{n-1})$$

word sequence	○ ○
left-to-right	● ● ● ● ● ● ● ● ● ● ● □ . . . .
bidir. (BERT 2018)	● ● ● ● ● ● ● ● ● ● ● □ ● ● ● ● ● ● ● ● ●

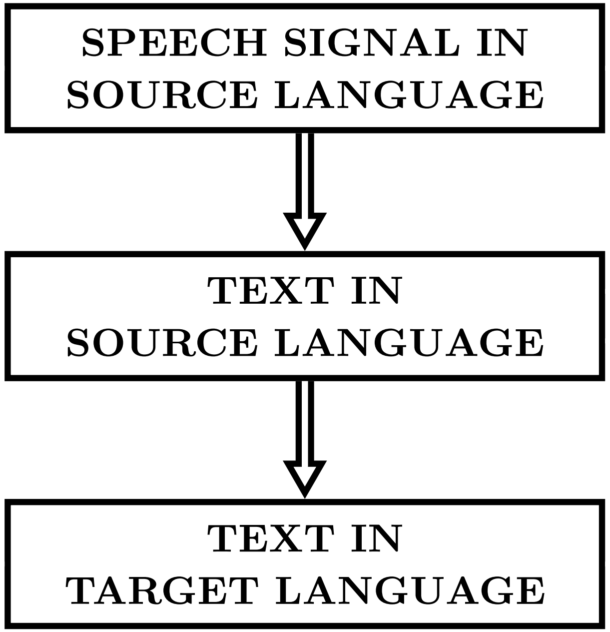
Speech Translation

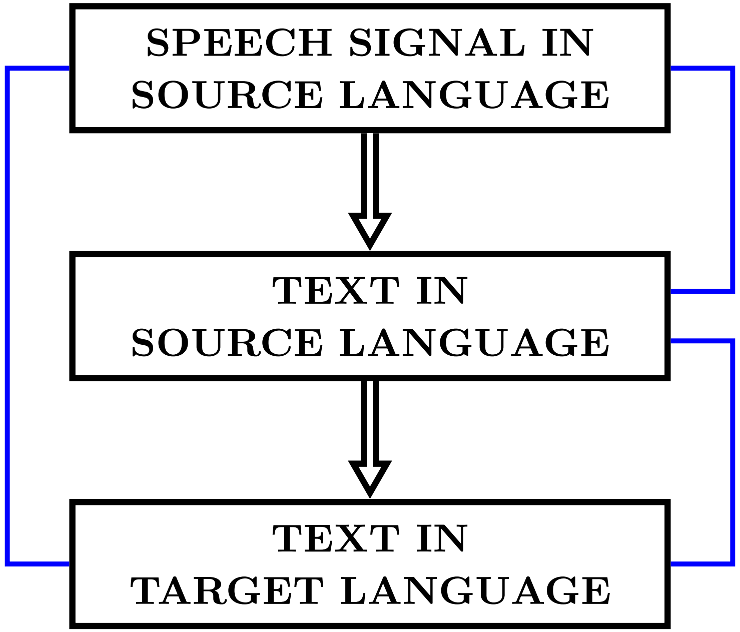


source audio X $\rightarrow$ source text F $\rightarrow$ target text E

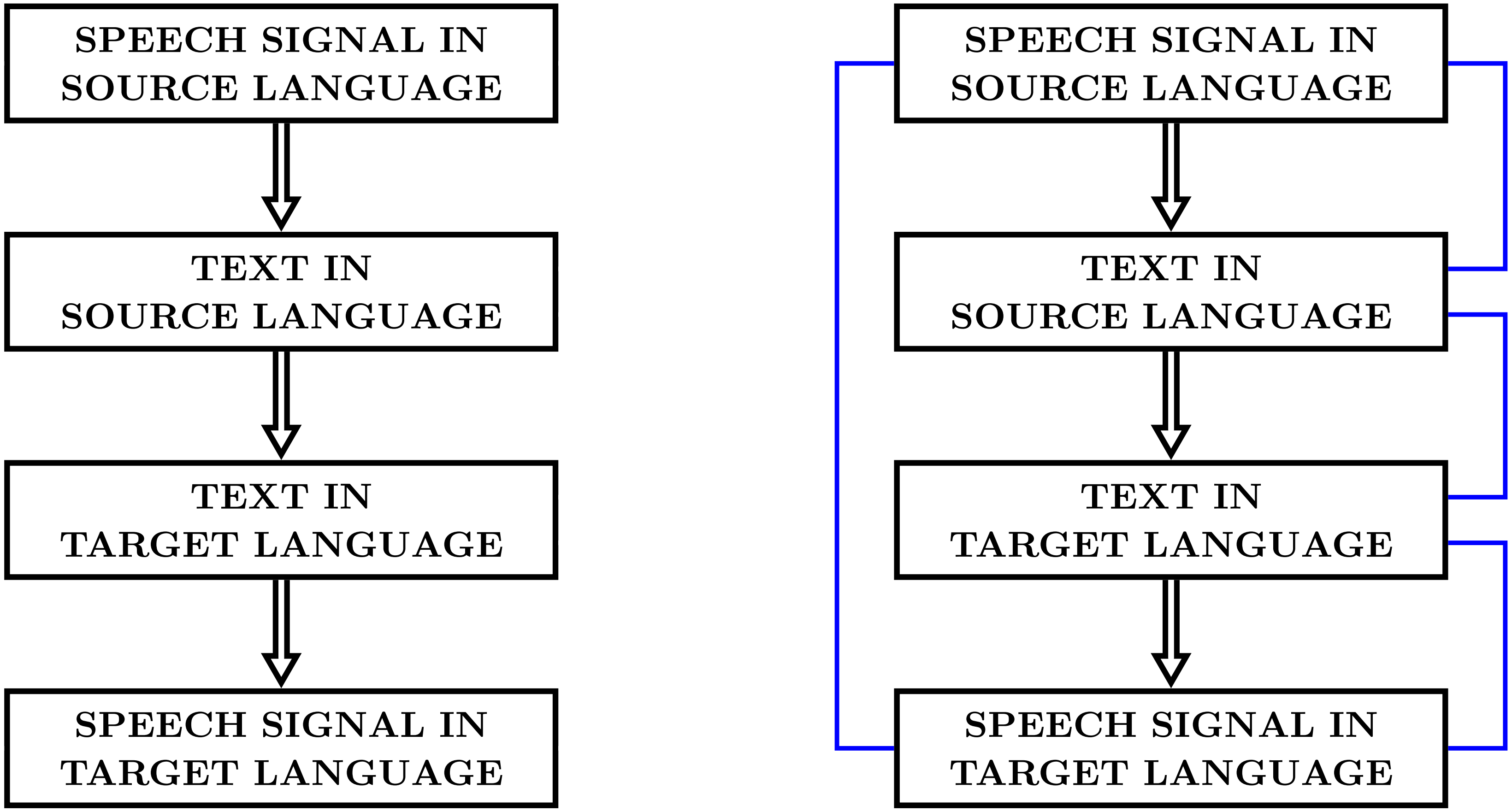
challenge: exploit three types of training data
– text MT: (F, E) sentence pairs (e. g. 100 Mio = 1-2 Bio words)
– ASR: (X, F) pairs (e. g. 5000 hours = 50 Mio words)
– speech-text MT: (X, E) (e. g. 1000 hours?)

Tasks in Human Language Technology: Speech-to-Text (Speech Translation)





Tasks in Human Language Technology: Speech-to-Speech Translation



ANN with Softmax Output

ANN: probabilistic interpretation:

- **ANN outputs [Bourlard & Wellekens 89]: class posteriors**
- **softmax [Bridle 89]: softmax = posterior of (class prior + Gaussian)**
  (assuming class-independent covariance matrix)

interpretation:

ANN with softmax = posterior of (class prior + Gaussian) + feature extraction

- **hidden layers perform feature extraction:**

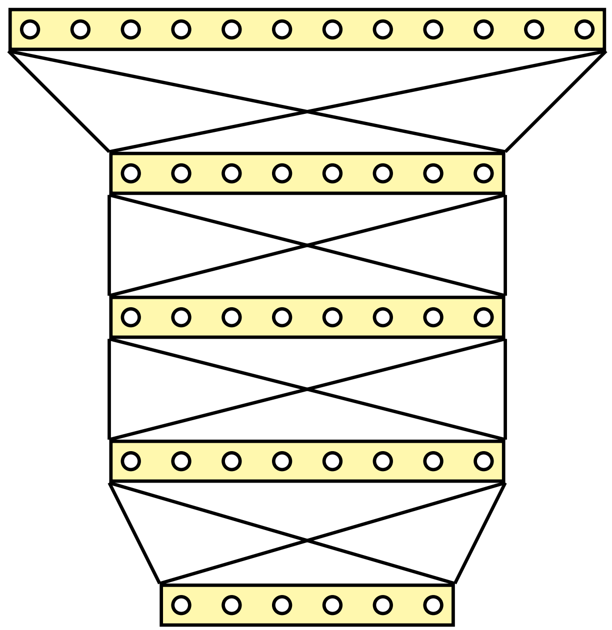
$$z \rightarrow x = f(z)$$

with feature vector $x \in \mathbb{R}^D$ before output layer
note: no dependence on class labels $c = 1, ..., C$

- **output layer: probability distribution over classes c**

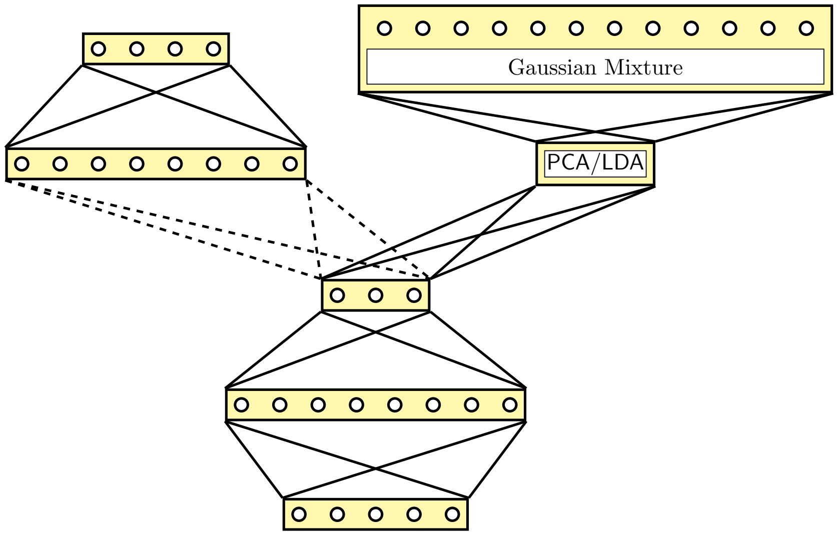
$$p(c|x) = \frac{\exp(\alpha_c + \lambda_c^t \cdot x)}{\sum_{c'} \exp(\alpha_{c'} + \lambda_{c'}^t \cdot x)}$$

with output layer weights $\lambda_c \in \mathbb{R}^D$
and offsets (biases) $\alpha_c \in \mathbb{R}$

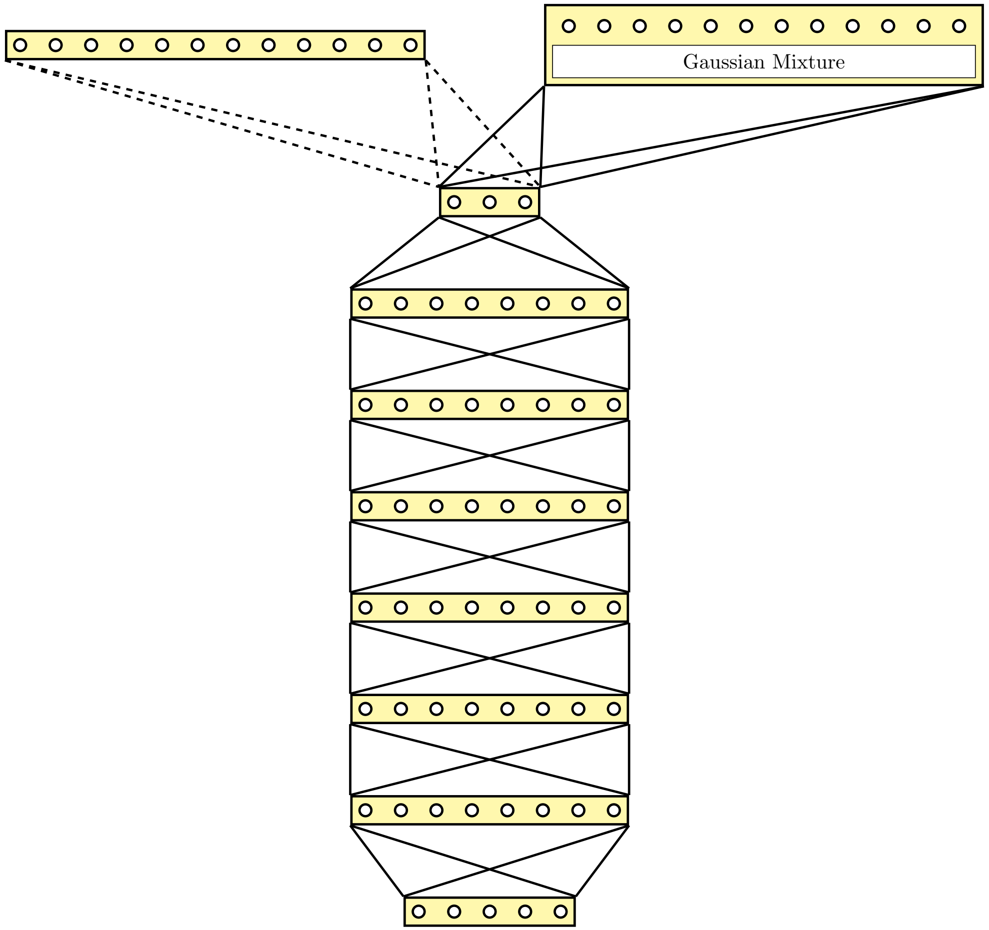


Tandem Approach: Explicit Feature Extraction

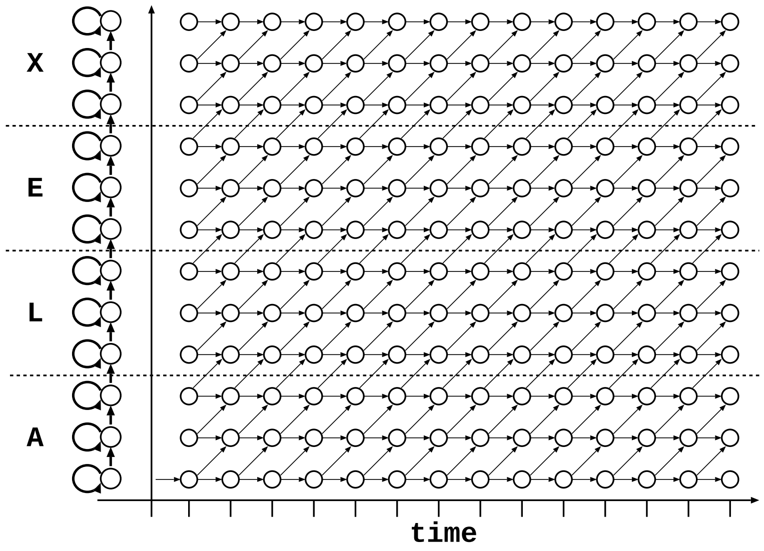
- **tandem approach: two parts:**
  MLP for feature extraction + generative HMM
  [Fontaine & Ris+ 97, Hermansky & Ellis+ 00]
- **extensions, e. g. bottleneck concept**
  [Stolcke & Grezl+ 06, Grezl & Fousek 08],
  [Valente & Vepa+ 07, Tüske & Plahl+ 11]

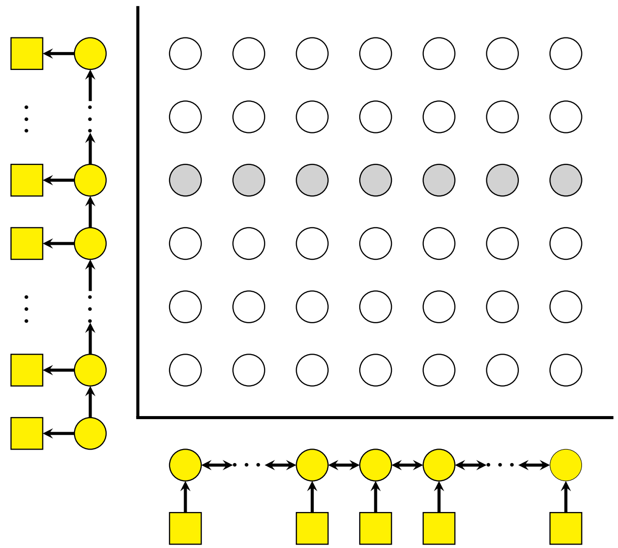


RWTH's Tandem Structure
[Tüske & Plahl+ 11]



Frame Label Posterior Probability





key quantity:
frame label posterior at time t
over labels $a = a_s$ for state/segment s:

$$q_t(a_s|x_1^T) \equiv q(y_t = a_s|h_t(x_1^T))$$

with frame labels $y_t, \ t = 1, ..., T$

acoustic encoder / feature extraction:
– deep MLP with window around t: $x_{t-\delta}^{t+\delta}$
– bi-direct. (LSTM) RNN: full context x_1^T
– transformer and conformer

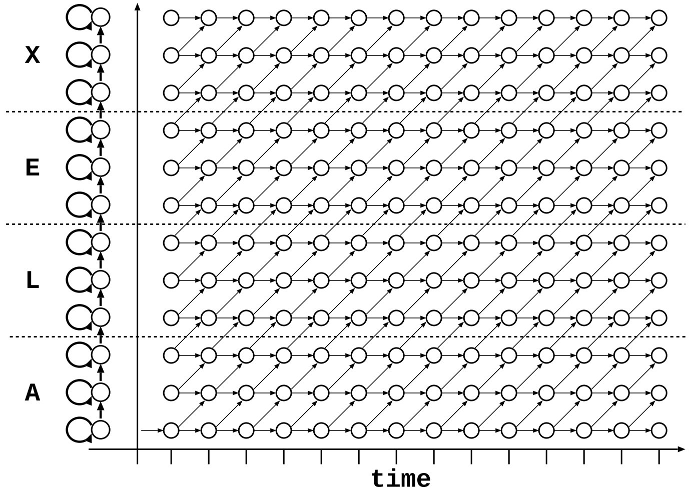
note: huge progress 1990-2020

label posteriors	$\circ$	$\circ$	...	$\circ$	$q(a\|h_t)$	$\circ$	...	$\circ$	$\circ$
	|	|	|	|	|	|	|	|	|
features	h_1	h_2	...	h_{t-1}	h_t	h_{t+1}	...	h_{T-1}	h_T
	|	|	|	|	|	|	|	|	|
acoustic vectors	x_1	x_2	...	x_{t-1}	x_t	x_{t+1}	...	x_{T-1}	x_T

Posterior HMM: From Hybrid HMM to CTC to RNN-T

direct re-writing of posterior HMM probability:

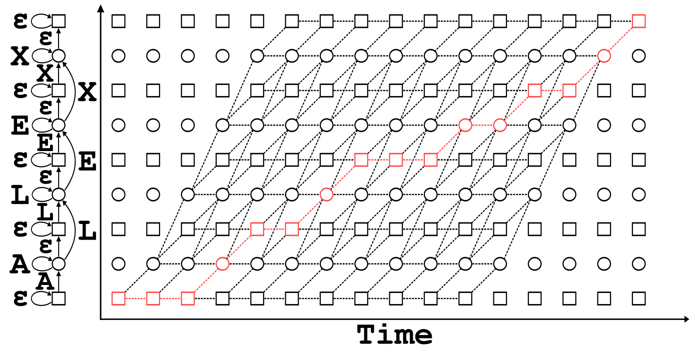
$$
\begin{aligned}
q_\vartheta(W = a_1^S | x_1^T) &= \sum_{s_1^T} \prod_t q_\vartheta(s_1^T, a_1^S | x_1^T) \\
&= \sum_{s_1^T} \prod_t q_\vartheta(s_{t+1}, y_t = a_{s_t} | s_t, a_{s_t - 1}, x_1^T) \\
&= \sum_{s_1^T} \prod_t q_\vartheta(s_{t+1} | s_t, a_{s_t}) \cdot q_\vartheta(y_t = a_{s_t} | a_{s_t - 1}, x_1^T)
\end{aligned}
$$



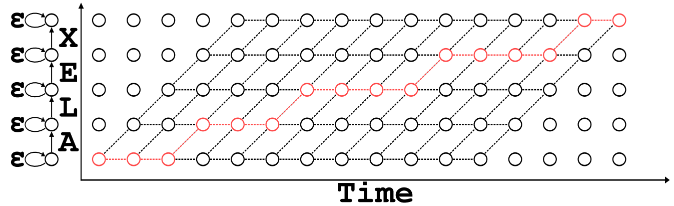
papers by RWTH: [Raissi & Beck$^+$ 20/21/22 arxiv]
[Zhou & Berger$^+$ 2021], [Zhou & Zeyer$^+$ 2021]

posterior HMM with ϵ symbol: CTC and transducer (RNN-T/RNN-A)
[Graves & Fernandez$^+$ 06, Graves 12, Sak & Shannon$^+$ 17]:
- **remove transition probabilities**
  and add special symbol: blank or ϵ:

$$\sum_{y_t \in \{a_s\} \cup \epsilon} q_\vartheta(y_t | a_{s'}, x_1^T) = 1$$



- **interpretation as probability of symbol repetition**
  and segmental model [Zhou & Zeyer$^+$ 2021]
- **transducer variant: no internal LM [Zhou & Berger$^+$ 2021]**

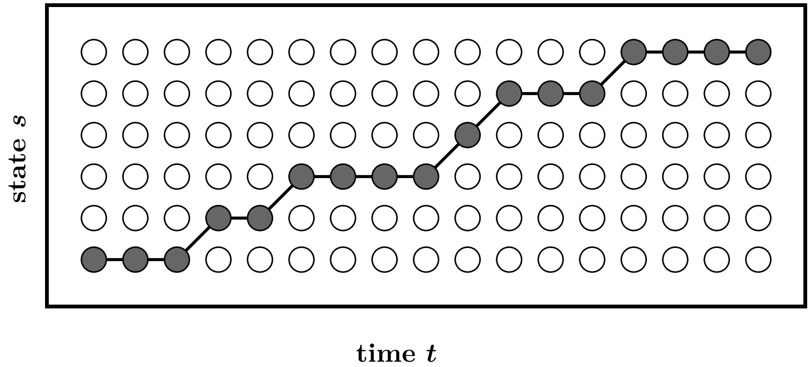


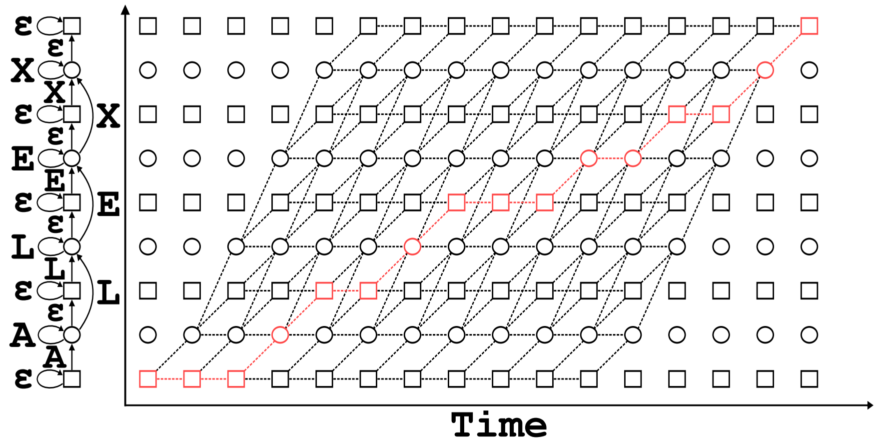
unifying principles
for posterior HMM, CTC and transducer with no internal LM:
- **hidden variable: alignment path**
- **sum criterion (or best path) along with EM-style training**
- **acoustic encoder to be included**

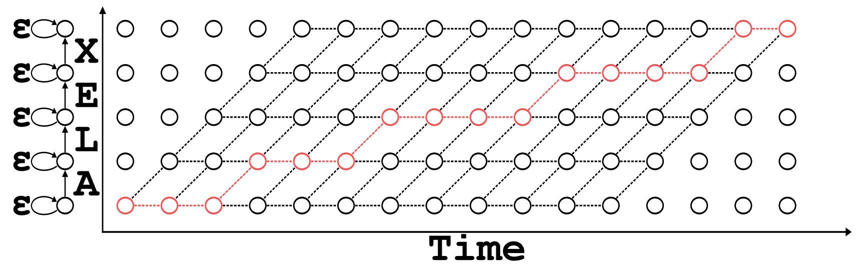
From Hybrid HMM to CTC to RNN-T

principal considerations:
- with ϵ/blank or transition prob.
- use of frame label priors
- duration constraints
- acoustic context dependence of labels: monophone, triphone, CART labels
- LM context in output generation: recursive, limited, none

practical tricks (maybe important):
- chunking
- spec-augment
- label smoothing
- extended training criteria: encoder loss, focal loss
- sub-sampling (e.g. 10→30→60 msec)
- ...







Tasks: Switchboard and Call Home

- **conversational speech: telephone speech, narrow band; challenging task: initial WER: 60% (and higher) on Switchboard**

- **training data for acoustic model: Switchboard corpus**
  - about 300 hours of speech
  - about 2400 two-sided recordings with an average of 200 seconds
  - 543 speakers

- **test set Hub5'00**
  - SWB: 20 telephone recordings form Switchboard studies
  - CHM: 20 telephone conversations from Call-Home US English Speech
  - total: 3.5 hours of speech

- **training data for language model**
  - vocabulary size fixed to 30k
  - Switchboard corpus: 2.9M running words
  - Fisher corpus: 21M running words

ASR: Results (April 2019) on Switchboard & Call Home

baseline models:
– language model: 4-gram count model
– acoustic model: hybrid HMM with CART (allophonic) labels:
LSTM bi-RNN with frame-wise cross-entropy training

– speaker/channel adaptation: i-vector [Dehak & Kenny^{+} 11]
– affine transformation [Gemello & Manai^{+} 06, Miao & Metze 15]

word error rates [%]:

adaptation	methods	SWB	CHM	average
no	baseline approach	9.7	19.1	14.4
	+ seq. discr. training (sMBR)	9.6	18.3	13.9
	+ LSTM-RNN language model	7.7	15.8	11.7
yes (i-vector)	baseline approach	9.0	18.0	13.5
	+ seq. discr. training (sMBR)	8.4	17.2	12.8
	+ LSTM-RNN language model	6.8	15.1	10.9
+ adaptation by affine transformation		6.7	13.5	10.2

overall improvements over baseline:
– 33% relative reduction in WER
– by seq. discr. training, LSTM-RNN language model and adaptation

Best Results on Call Home (CHM) and Switchboard (SWB)
(best word error rates [%] reported)

team	CHM	SWB	training data, remarks
Johns Hopkins U 2017	18.1	9.0	300h, no ANN-LM, single model, data perturbation
Microsoft 2017	17.7	8.2	300h, ResNet, with ANN-LM
ITMO U 2016	16.0	7.8	300h, with ANN-LM, model comb., data perturbation
Google 2019/arXiv	14.1	6.8	300h, attention models
RWTH U 2017	15.7	8.2	300h, with ANN-LM, model comb.
RWTH U 2019/arXiv	13.5	6.7	300h, single models, adaptation
Microsoft 2017	12.0	6.2	2000h, model comb.
IBM 2017	10.0	5.5	2000h, model comb.
Capio 2017	9.1	5.0	2000h, model comb.

ASR: Librispeech Task: Hybrid HMM vs. Attention (Vassil Panayotov & Daniel Povey)

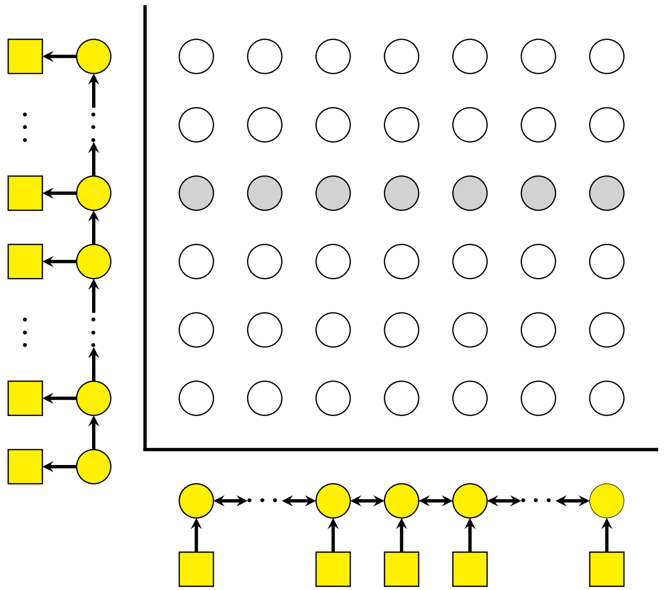
speech data: read audiobooks from the LibriVox project
with training data:
– acoustic model: 960 hrs of speech
– language model: 800 Mio words

word error rates[%]:

		dev		test	
team	approach	1st half	2nd half	1st half	2nd half
Irie, Zeyer et al. RWTH (Interspeech 2019)	attention with BPE units, 'no' LM	4.3	12.9	4.4	13.5
	+ LSTM-RNN LM	3.0	9.1	3.5	10.0
	+ transformer LM	2.9	8.8	3.1	9.8
Lüscher, Beck et al. RWTH (Interspeech 2019)	hybrid HMM, CART, 4g LM	4.3	10.0	4.8	10.7
	+ seq. disc. training	3.7	8.7	4.2	9.3
	+ LSTM-RNN LM	2.4	5.8	2.8	6.2
	+ transformer LM	2.3	5.2	2.7	5.7

Zeghidour et al., FB 2018	gated CNN with letters/words	3.2	10.1	3.4	11.2
Irie et al., Google 2019	attention with WPM units	3.3	10.3	3.6	10.3
Park et al., Google 2019	attention ... data augmentation	-	-	2.5	5.8

common properties:
– input: acoustic encoder: representation/state vectors $h_t = h_t(x_1^T), t = 1, ..., T$
– output: (phoneme) labels $a_s,\ s = 1, ..., S$ with/without integrated language model



- **attention: averaging over internal representations h_t:**

$$p(a_1^S|x_1^T) \;=\; \prod_s p(a_s|a_0^{s-1}, x_1^T) = \prod_s p(a_s|a_{s-1}, r_{s-1}, c_s)$$

$$c_s \;:=\; \sum_t p(t|a_0^{s-1}, x_1^T) \cdot h_t$$

with context vector c_s and output state vector r_s

criticism for ASR: lack of strict monotonicity and localization

- **posterior HMM: summing over the products along the paths, i.e. models:**

$$p(a_1^S|x_1^T) \;=\; \sum_{s_1^T} \prod_t p\Big(s_{t+1}, y_t = a_{s_t} \Big| s_t, a_{s_t-1}, h_t\Big)$$

$$= \sum_{s_1^T} \exp\Big[\sum_t \log\, p\Big(s_{t+1}, y_t = a_{s_t} \Big| s_t, a_{s_t-1}, h_t\Big)\Big]$$

results on phoneme/grapheme RNN-Transducer (RNN-T):
IBM research [Saon & Tüske+ 2021] and RWTH [Zhou & Berger+ 2021]

table and results from [Saon & Tüske+ 2021]
on Switchboard (SWB) and Call-Home (CHM):

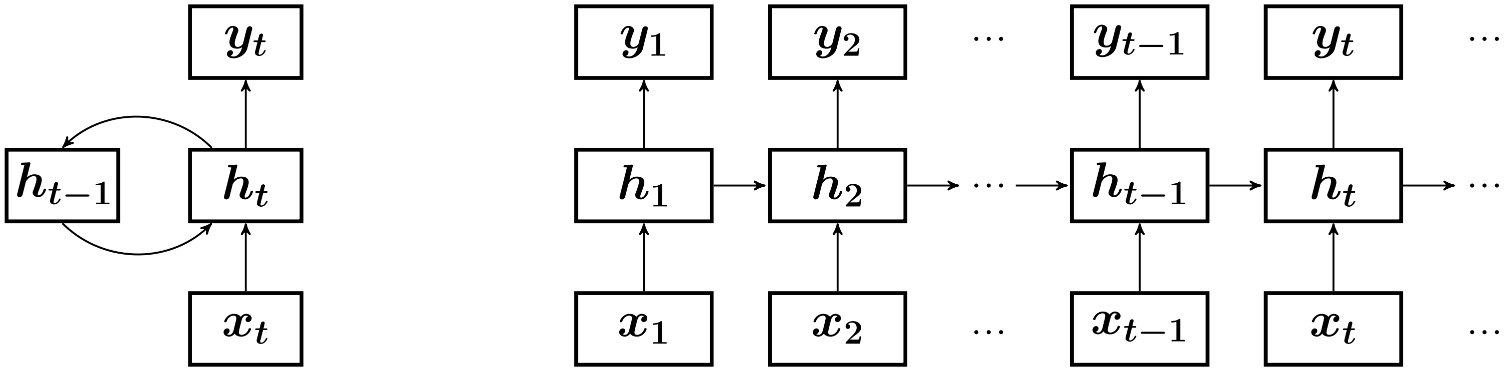
authors	team	approach		WER[%]	
		acoust.model	lang.model	SWB	CHM
Saon & Tüske+ 2021	IBM	RNN-T	LSTM-RNN	6.3	13.1
Tüske & Saon+ 2020	IBM	attention	LSTM-RNN	6.4	12.5
Park & Chan+ 2019	Google	attention	LSTM-RNN	6.8	14.1
Hadiani & Sameti+ 2018	JHU	latt.free MMI	RNN	7.5	14.6
Irie & Zeyer+ 2019	RWTH	hybrid HMM	transformer	6.7	12.9

more results on Italian and Spanish (conversational telephone speech)

conclusions based on [Saon & Tüske+ 2021, Zhou & Berger+ 2021]:
similar performance like hybrid HMM

RWTHAACHEN UNIVERSITY

ASR: sequence-to-sequence processing



from simple ANN to RNN:
– introduce a memory (or context) component to keep track of history
– result: two types of input at time t: memory h_{t-1} and observation x_t

extensions:

- **(succesful!) application to ASR:**
  [Robinson 94]
- **bidirectional structure [Schuster & Paliwal 97]**
- **LSTM: long short-term memory**
  [Hochreiter & Schmidhuber 97, Gers & Schraudolph^{+} 02]

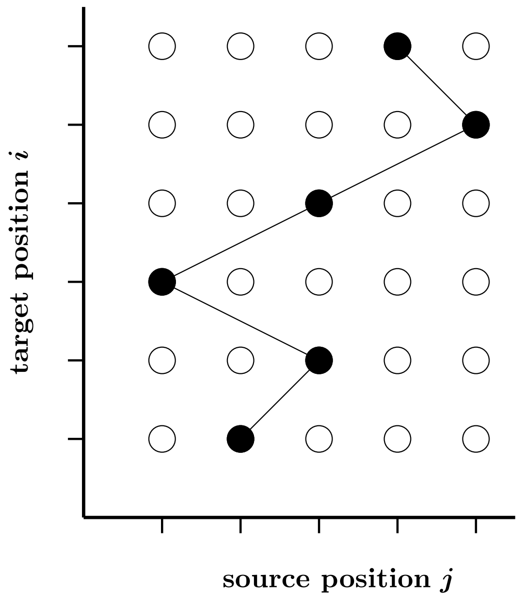
AppTek

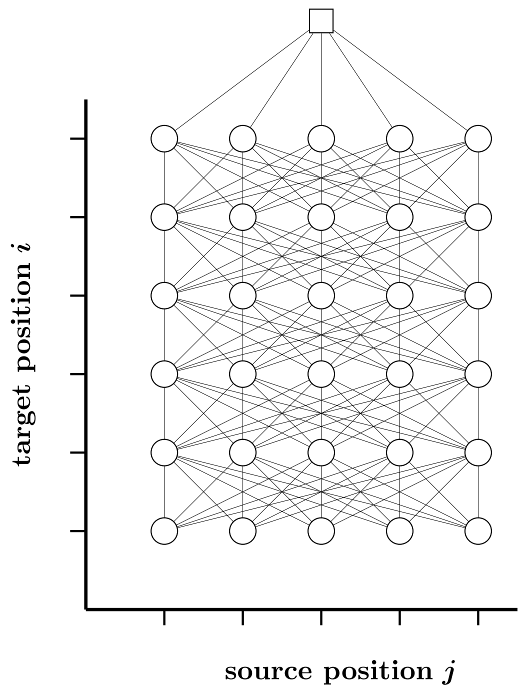
Machine Translation: Neural HMM

- **translation: from source sentence $f_1^J = f_1...f_j...f_J$ to target sentence $e_1^I = e_1...e_i...e_I$**
- **alignment direction: from target to source: $i \rightarrow j = b_i$**
- **first-order hidden alignments and factorization:**

$$p(e_1^I|f_1^J) = \sum_{b_1^I} p(b_1^I, e_1^I|f_1^J) = \sum_{b_1^I} \prod_i p(b_i, e_i|b_{i-1}, e_0^{i-1}, f_1^J)$$

- **resulting model: exploit first-order structure (or zero-order)**
  training: backpropagation within EM algorithm





Experimental Results

- **WMT task: German → English:**
  - **training data: 6M sentence pairs = (137M, 144M) words**
  - **test data: (about) 3k sentence pairs = (64k, 67k) words**
- **WMT task: Chinese → English:**
  - **training data: 14M sentence pairs = (920M Chinese letters, 364M English words)**
  - **test data: (about) 2k sentence pairs = (153k Chinese letters, 71k English words)**
- **performance measures:**
  - **BLEU [%]: accuracy measure: "the higher, the better"**
  - **TER [%]: error measure: "the lower, the better"**
- **basic units for implementation:**
  - **BPE (*byte pair encoding*) units rather than full-form words**
  - **alphabet size: about 40k**
- **RWTH papers (with preliminary results):**
  [Wang & Alkhouli$^+$ 17, Wang & Zhu$^+$ 18]

Comparison: Best Results

	German→English				Chinese→English			
	test2017		test2018		dev2017		test2017	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
LSTM-RNN attention	32.1	56.3	38.8	48.1	21.4	63.6	22.9	62.0
self-attention transformer	33.4	55.3	40.4	46.8	21.8	62.9	23.5	60.1
neural HMM	31.9	56.6	38.3	48.3	20.8	63.2	22.4	61.4

conclusions about neural HMM:
– (nearly) competitive with LSTM-RNN attention approach
– some performance gap to self-attention approach

– room for improvement of neural HMM

RWTH AACHEN UNIVERSITY

- **LSTM-RNN based representations for input and output:**
  4 layers of encoder and 1 layer of decoder
- **independent models of alignment and lexicon**
  (no parameter sharing as in attention approach)

HMM	German→English						Chinese→English					
	#Par	PPL	test2017		test2018		#Par	PPL	dev2017		test2017	
			BLEU	TER	BLEU	TER			BLEU	TER	BLEU	TER
zero-order	129M	5.29	30.9	57.4	37.4	48.9	125M	8.12	20.1	65.1	20.7	64.2
first-order	136M	4.64	31.6	56.5	38.7	48.4	138M	7.63	20.1	64.0	22.0	63.2

machine translation from source source to target language:

(source: foreign) $f_1^J \to e_1^I$ **(target: English)**

key concepts for modelling posterior probability $p(e_1^I|f_1^J)$

- **direct approach: use unidirectional RNN over target positions** $i = 1, ..., I$ **with internal state vector** s_i**:**

$$p(e_1^I|f_1^J) = \prod_i p(e_i|e_0^{i-1}, f_1^J) = \prod_i p(e_i|e_{i-1}, s_{i-1}, f_1^J)$$

  interpretation: extended language model for target word sequence

- **additional component: attention mechanism for localization**

$$p(e_i|e_{i-1}, s_{i-1}, f_1^J) = p(e_i|e_{i-1}, s_{i-1}, c_i)$$

  with a context vector: $c_i := C(s_{i-1}, f_1^J)$

word embeddings and representations:

- **word embedding for target sequence:**
  – word symbol: e_i
  – word vector: $\tilde{e}_i = R_e(e_i)$ with the embedding (matrix) R_e
- **word embedding for source sequence:**
  – word symbol: f_j
  – word vector: $\tilde{f}_j = R_f(f_j)$ with the embedding (matrix) R_f
- **word representation h_j for source sequence using a bidirectional RNN:** $h_j = H_j(f_1^J)$

warning:
– concept: clear distinction between $f_j, \tilde{f}_j, h_j$
– notation and terminology: not necessarily consistent

approach:

- **input: bidirectional RNN over source positions j: $f_1^J \rightarrow h_j = H_j(f_1^J)$**
- **output: unidirectional RNN over target positions i:**

$$y_i = Y(y_{i-1}, s_{i-1}, c_i)$$

conventional notation:

$$p(e_i|\tilde{e}_{i-1}, s_{i-1}, c_i)$$

with RNN state vector $s_i = S(s_{i-1}, \tilde{e}_i, c_i)$ and context vector $c_i = C(s_{i-1}, h_1^J)$

- **context vector c_i: weighted average of source word representations:**

$$c_i = \sum_j \alpha(j|i, s_{i-1}, h_1^J) \cdot h_j \qquad\qquad \alpha(j|i, s_{i-1}, h_1^J) = \frac{\exp(A[s_{i-1}, h_j])}{\sum_{j'} \exp(A[s_{i-1}, h_{j'}])}$$

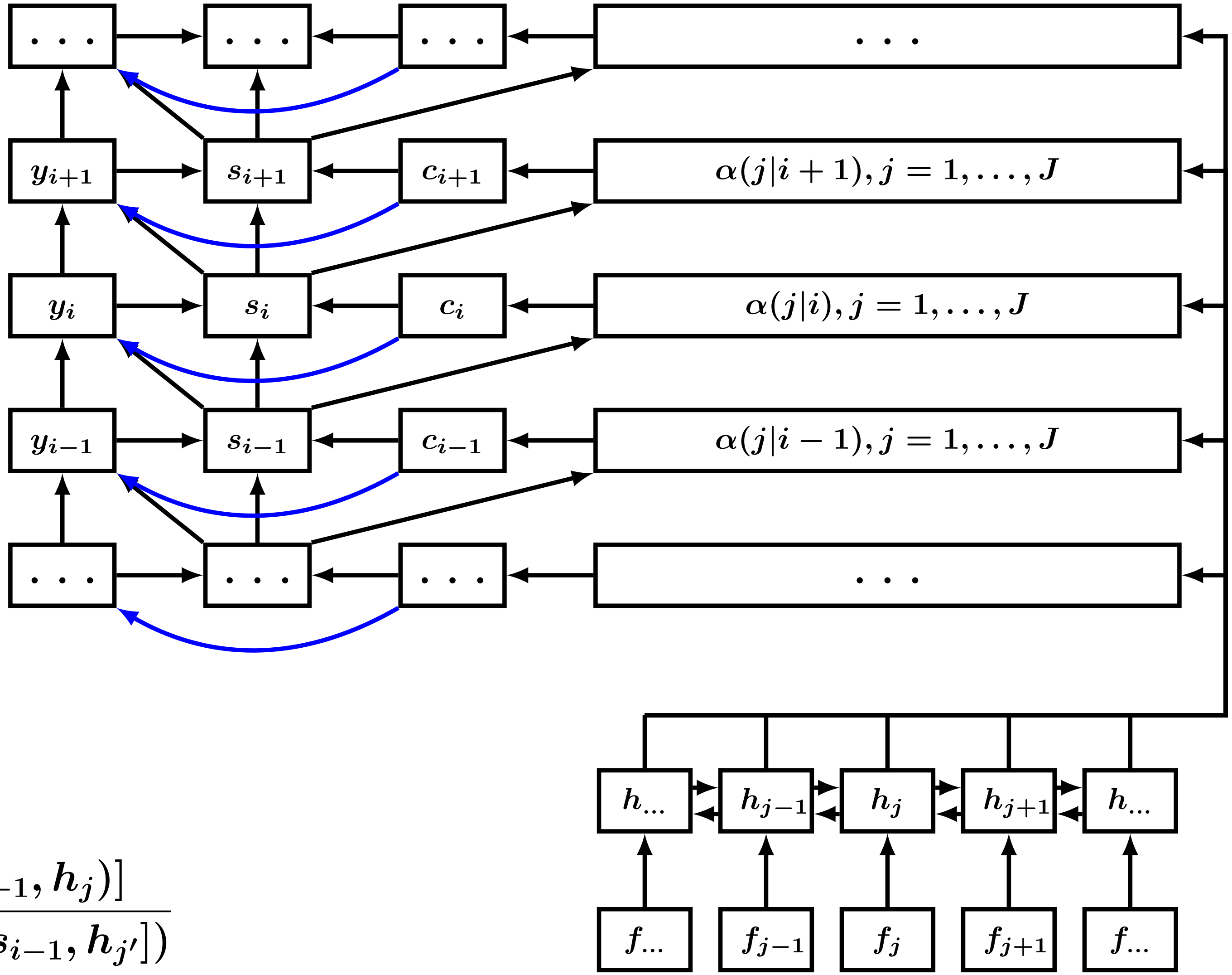
with the normalized attention weights $\alpha(j|i, s_{i-1}, h_1^J)$
and real-valued attention scores $A[s_{i-1}, h_j]$

State of the Art: Attention-based Neural MT [Bahdanau & Cho+ 15]

principle:

- **input: source sequence:**
  $f_1^J \to h_j = H_j(f_1^J)$
- **output distribution:**
  $y_i \equiv p_i(e|\tilde{e}_{i-1}, s_{i-1}, c_i)$
  notation in ANN style:
  $y_i = Y(y_{i-1}, s_{i-1}, c_i)$
- **state vector of target RNN:**
  $s_i = S(s_{i-1}, y_i, c_i)$
- **weighted context vector:**
  $c_i = \sum_j \alpha(j|i, s_{i-1}, h_1^J) \cdot h_j$
- **attention weights:**

$$\alpha(j|i, s_{i-1}, h_1^J) = \frac{\exp(A[s_{i-1}, h_j)]}{\sum_{j'} \exp(A[s_{i-1}, h_{j'}])}$$

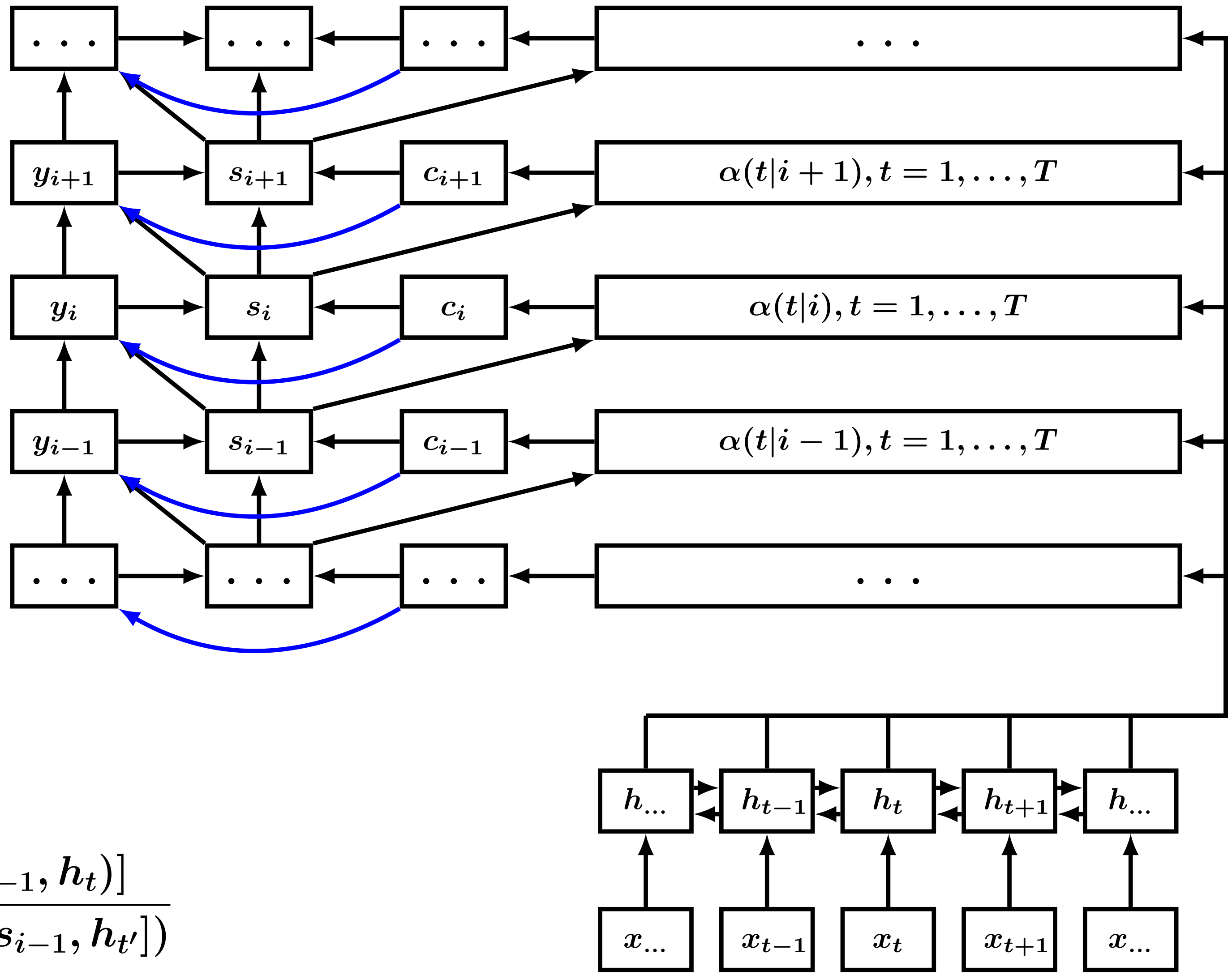


Attention-based ASR: $x_1^T \rightarrow a_1^I$ ([Bahdanau & Cho+ 15] for MT)

principle:

- **input: source sequence:**
  $(x_1^T, t) \rightarrow h_t = H_t(x_1^T)$
- **output distribution:**
  $y_i \equiv p_i(a|\tilde{a}_{i-1}, s_{i-1}, c_i)$
  notation in ANN style:
  $y_i = Y(y_{i-1}, s_{i-1}, c_i)$
- **state vector of target RNN:**
  $s_i = S(s_{i-1}, y_i, c_i)$
- **weighted context vector:**
  $c_i = \sum_t \alpha(t|i, s_{i-1}, h_1^T) \cdot h_t$
- **attention weights:**

$$\alpha(t|i, s_{i-1}, h_1^T) = \frac{\exp(A[s_{i-1}, h_t)]}{\sum_{t'} \exp(A[s_{i-1}, h_{t'}])}$$



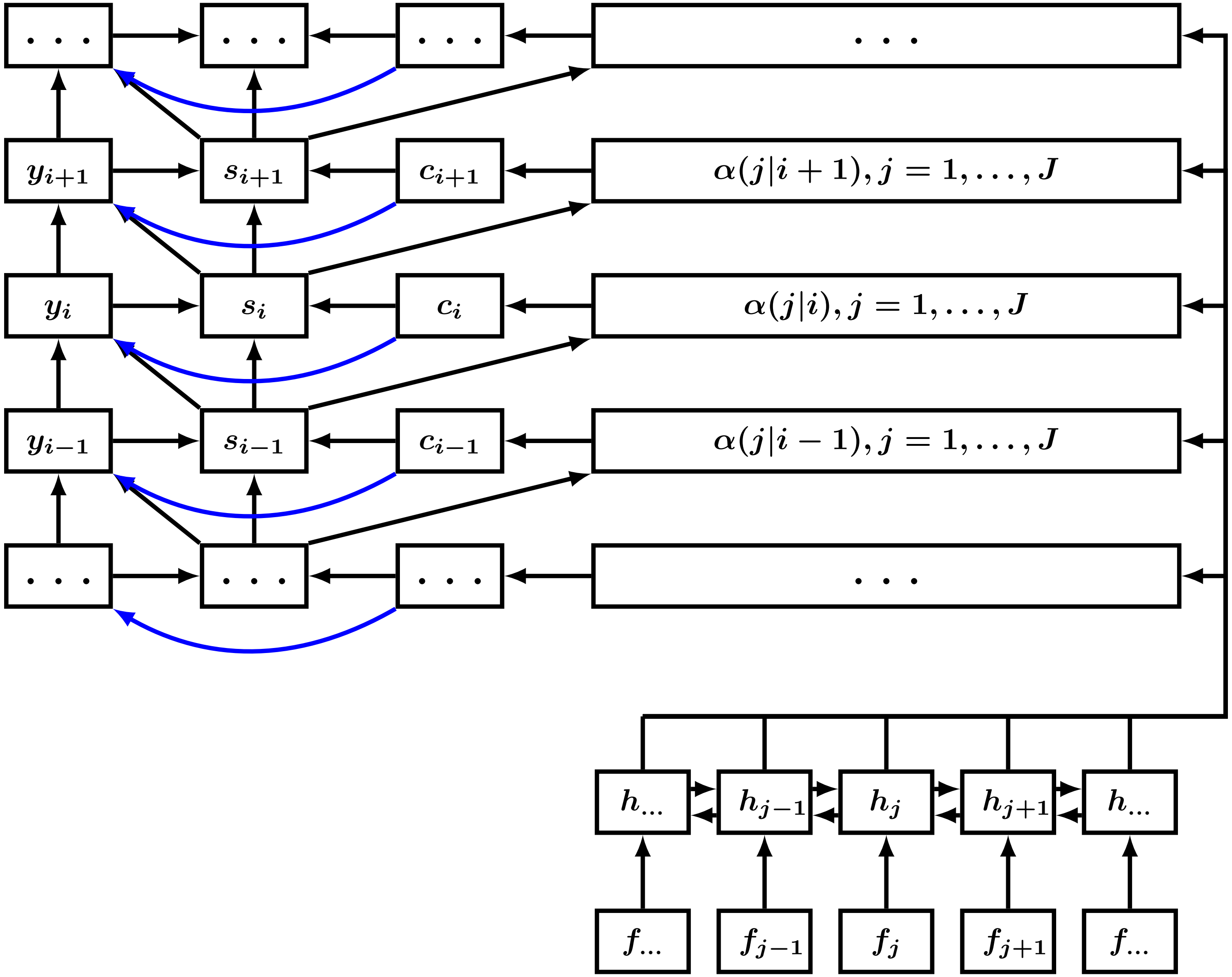
Attention-based Neural MT:
Sequential Order of Operations

preparations:

- **input preprocessing:**
  $f_1^J \rightarrow h_j = H_j(f_1^J)$
- **available at position $i-1$:**
  $\tilde{e}_{i-1} \equiv y_{i-1},\ s_{i-1},\ c_{i-1}$

sequence of operations for position i:

1. **attention weights:**
   $\alpha(j|i, s_{i-1}, h_1^J) = \ldots$
2. **context vector:**
   $c_i = \sum_j \alpha(j|i, s_{i-1}, h_1^J) \cdot h_j$
3. **output distribution:**
   $y_i = Y(y_{i-1}, s_{i-1}, c_i)$
4. **state vector:**
   $s_i = S(s_{i-1}, y_i, c_i)$



Attention Weights
Feedforward ANN vs. Dot Product

re-consider attention weights:

$$\alpha(j|i, s_{i-1}, h_1^J) = \frac{\exp(A[s_{i-1}, h_j])}{\sum_{j'} \exp(A[s_{i-1}, h_{j'}])}$$

two approaches to modelling attention scores $A[s_{i-1}, h_j]$:

- **additive variant: feedforward (FF) ANN:**

$$A[s_{i-1}, h_j] := v^T \cdot \tanh(S s_{i-1} + H h_j)$$

  with matrices S and H and vector v
  basic implementation: one FF layer + softmax

- **multiplicative variant: (generalized) dot product between vectors:**

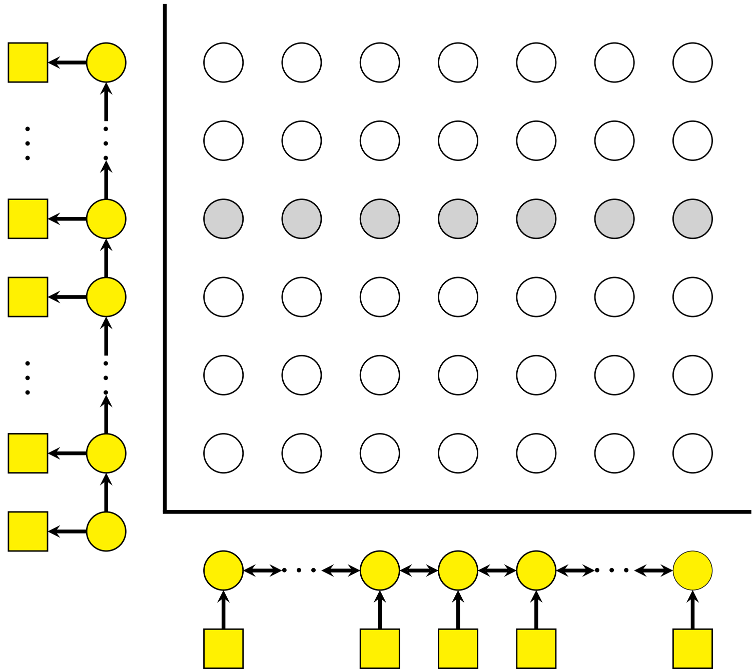
$$A[s_{i-1}, h_j] := s_{i-1}^T \cdot W \cdot h_j$$

  with a attention matrix W

experimental result: not much difference

common properties in both approaches:

– bi-directional LSTM RNN over input words f_j, $j = 1, ..., J$
– uni-directional LSTM RNN over output words e_i, $i = 1, ..., I$



- direct HMM (finite-state model): summing over probability models

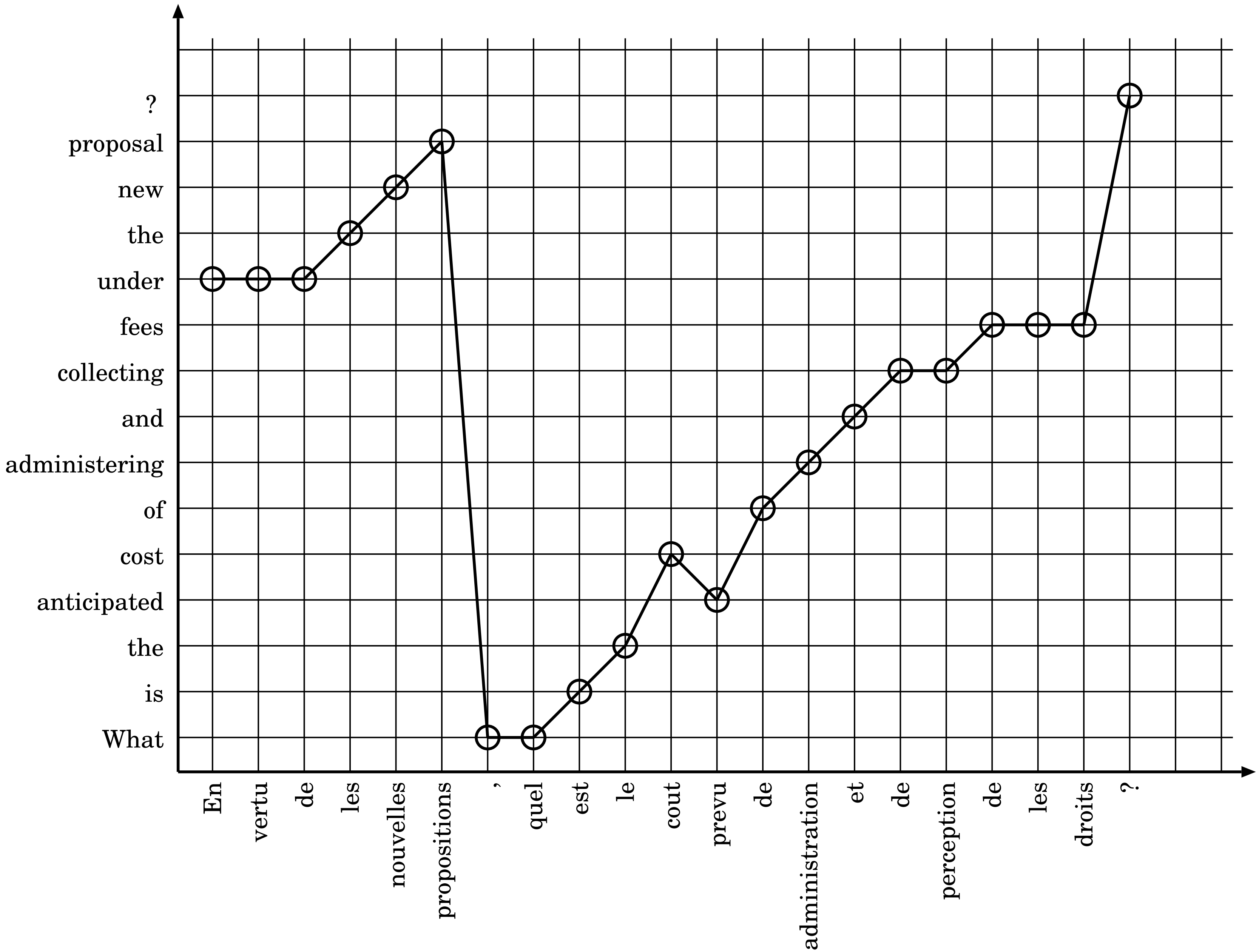
$$p(e_1^I | f_1^J) = \sum_{b_1^I} \prod_i p(b_i, e_i | b_{i-1}, e_0^{i-1}, f_1^J)$$

- attention mechanism: averaging over internal RNN representations h_j:

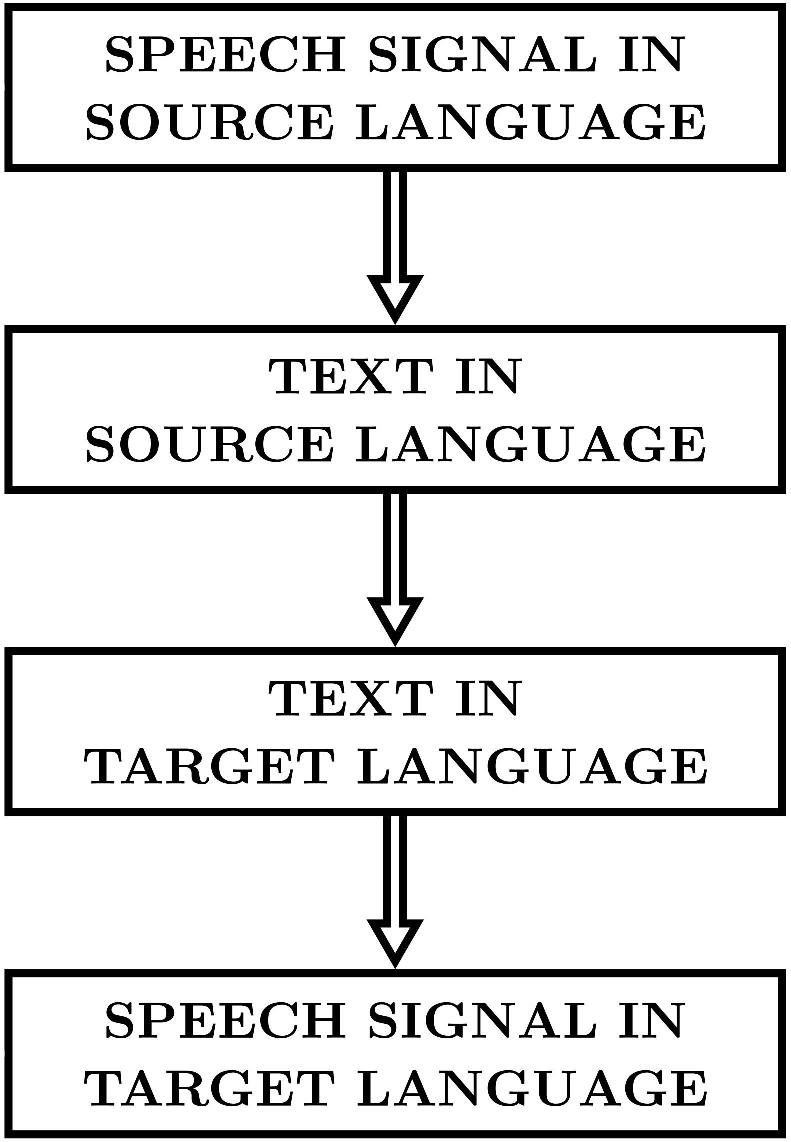
$$p(e_i | e_0^{i-1}, f_1^J) = p(e_i | e_{i-1}, s_{i-1}, c_i)$$

$$\text{with} \quad c_i = \sum_j p(j | e_0^{i-1}, f_1^J) \cdot h_j(f_1^J)$$

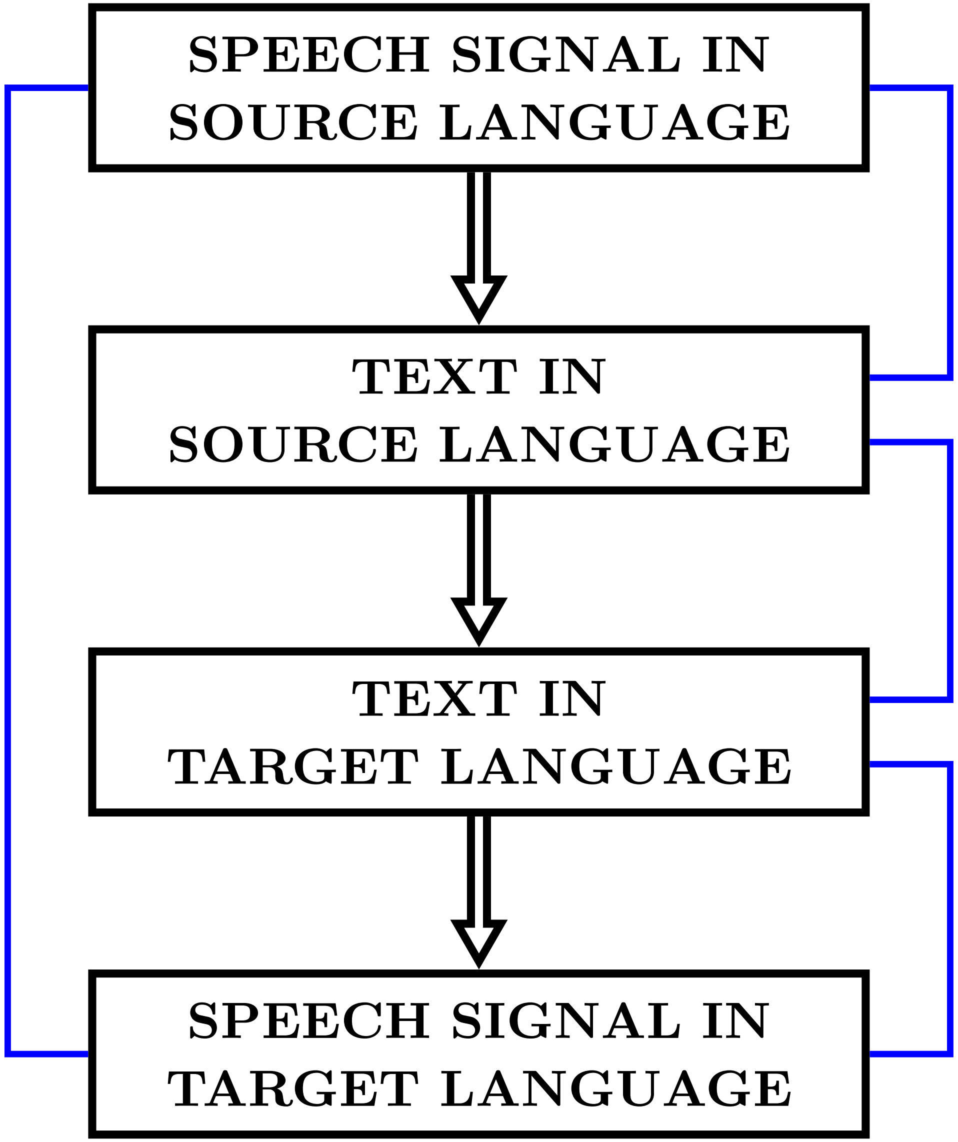
Word Alignments (based on HMM)
(learned automatically; Canadian Parliament)



Tasks in Human Language Technology: Speech-to-Speech Translation



Tasks in Human Language Technology: Speech-to-Speech Translation



History:

- **1989 [Nakamura & Shikano 89]:**
  English word category prediction based on neural networks.
- **1993 [Castano & Vidal$^+$ 93]:**
  Inference of stochastic regular languages through simple recurrent networks
- **2000 [Bengio & Ducharme$^+$ 00]:**
  A neural probabilistic language model
- **2007 [Schwenk 07]: Continuous space language models**
  2007 [Schwenk & Costa-jussa$^+$ 07]: Smooth bilingual n-gram translation (!)
- **2010 [Mikolov & Karafiat$^+$ 10]:**
  Recurrent neural network based language model
- **2012 RWTH Aachen [Sundermeyer & Schlüter$^+$ 12]:**
  LSTM recurrent neural networks for language modeling

today: ANNs in language show competitive results.

History of ANN based approaches to MT:

- **1997 [Neco & Forcada 97]:**
  asynchronous translations with recurrent neural nets
- **1997 [Castano & Casacuberta 97, Castano & Casacuberta^{+} 97]:**
  machine translation using neural networks and finite-state models
- **2007 [Schwenk & Costa-jussa^{+} 07]:**
  smooth bilingual n-gram translation
- **2012 [Le & Allauzen^{+} 12, Schwenk 12]:**
  continuous space translation models with neural networks
- **2014 [Devlin & Zbib^{+} 14]:**
  fast and robust neural networks for SMT
- **2014 [Sundermeyer & Alkhouli^{+} 14]:**
  recurrent bi-directional LSTM RNN for SMT
- **2015 [Bahdanau & Cho^{+} 15]:**
  joint learning to align and translate

6 References

[Baevski & Schneider$^+$ 20] A. Baevski, S. Schneider, M. Auli: VQ-Wav2Vec: Self-Supervised Learning of Discrete Speech Representations. Facebook AI Research, Menlo Park, CA, arxiv, 16-Feb-2021.

[Bahdanau & Cho$^+$ 15] D. Bahdanau, K. Cho, Y. Bengio: Neural machine translation by jointly learning to align and translate. Int. Conf. on Learning and Representation (ICLR), San Diego, CA, May 2015.

[Bahl & Jelinek$^+$ 83] L. R. Bahl, F. Jelinek, R. L. Mercer: A Maximum Likelihood Approach to Continuous Speech Recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 5, pp. 179-190, March 1983.

[Bahl & Brown$^+$ 86] L. R. Bahl, P. F. Brown, P. V. de Souza, R. L. Mercer: Maximum mutual information estimation of hidden Markov parameters for speech recognition. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Tokyo, pp.49-52, April 1986.

[Beck & Schlüter$^+$ 15] E. Beck, R. Schlüter, H. Ney: Error Bounds for Context Reduction and Feature Omission, Interspeech, Dresden, Germany, Sep. 2015.

[Bang & Cahyawijaya$^+$ 23] Y. Bang, S. Cahyawijaya, N. Lee et al.: A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. HKUST, arxiv, 28-Feb-2023.

[Bengio & Ducharme$^+$ 00] Y. Bengio, R. Ducharme, P. Vincent: A neural probabilistic language model. Advances in Neural Information Processing Systems (NIPS), pp. 933-938, Denver, CO, USA, Nov. 2000.

[Botros & Irie$^+$ 15] R. Botros, K. Irie, M. Sundermeyer, H. Ney: On Efficient Training of Word Classes and Their Application to Recurrent Neural Network Language Models. Interspeech, pp.1443-1447, Dresden, Germany, Sep. 2015.

[Bourlard & Wellekens 87] H. Bourlard, C. J. Wellekens: Multilayer perceptrons and automatic speech recognition. First Int. Conf. on Neural Networks, pp. 407-416, San Diego, CA, 1987.

[Bourlard & Wellekens 89] H. Bourlard, C. J. Wellekens: 'Links between Markov Models and Multilayer Perceptrons', in D.S. Touretzky (ed.): "Advances in Neural Information Processing Systems I", Morgan Kaufmann Pub., San Mateo, CA, pp.502-507, 1989.

[Bridle 82] J. S. Bridle, M. D. Brown, R. M. Chamberlain: An Algorithm for Connected Word Recognition. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, pp. 899-902, May 1982.

[Bridle 89] J. S. Bridle: Probabilistic Interpretation of Feedforward Classification Network Outputs with Relationships to Statistical Pattern Recognition, in F. Fogelman-Soulie, J. Herault (eds.): 'Neuro-computing: Algorithms, Architectures and Applications', NATO ASI Series in Systems and Computer Science, Springer, New York, 1989.

[Bridle & Dodd 91] J. S. Bridle, L. Dodd: An Alphanet Approach To Optimising Input Transformations for Continuous Speech Recognition. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Toronot, pp. 277-280, April 1991.

[Brown & Della Pietra$^+$ 93] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, R. L. Mercer: Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics, Vol. 19.2, pp. 263-311, June 1993.

[Brown & Mann$^+$ 22] T. R. Brown, B. Mann, N. Ryder et al.: Language Models are Few-Shot Learners. OpenAI (GPT-3), arxiv, 22-Jul-2022.

[Collobert & Weston 08] R. Collobert, J. Weston: A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. Int. Conference on Machine Learning (ICML), 2008.

[Collobert & Weston$^+$ 11] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa: Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research, 2011.

[Castano & Vidal$^+$ 93] M.A. Castano, E. Vidal, F. Casacuberta: Inference of stochastic regular languages through simple recurrent networks. IEE Colloquium on Grammatical Inference: Theory, Applications and Alternatives, pp. 16/1-6, Colchester, UK, April 1993.

[Castano & Casacuberta 97] M. Castano, F. Casacuberta: A connectionist approach to machine translation. European Conf. on Speech Communication and Technology (Eurospeech), pp. 91–94, Rhodes, Greece, Sep. 1997.

[Castano & Casacuberta^{+} 97] M. Castano, F. Casacuberta, E. Vidal: Machine translation using neural networks and finite-state models. Int. Conf. on Theoretical and Methodological Issues in Machine Translation (TMI), pp. 160-167, Santa Fe, NM, USA, July 1997.

[Dahl & Ranzato^{+} 10] G. E. Dahl, M. Ranzato, A. Mohamed, G. E. Hinton: Phone recognition with the mean-covariance restricted Boltzmann machine. Advances in Neural Information Processing Systems (NIPS) 23, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, Eds. Cambridge, MA, MIT Press, 2010, pp. 469-477.

[Dahl & Yu^{+} 12] G. E. Dahl, D. Yu, L. Deng, A. Acero: Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. IEEE Tran. on Audio, Speech and Language Processing, Vol. 20, No. 1, pp. 30-42, Jan. 2012.

[Dehak & Kenny^{+} 11] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, P. Ouellet: Front-End Factor Analysis for Speaker Verification IEEE Trans. on audio, speech, and language processing, pp. 788-798, Vol. 19, No. 4, May 2011.

[Devlin & Zbib^{+} 14] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, J. Makhoul: Fast and Robust Neural Network Joint Models for Statistical Machine Translation. Annual Meeting of the ACL, pp. 1370–1380, Baltimore, MA, June 2014.

[Doetsch & Hannemann^{+} 17] P. Doetsch , M. Hannemann, R. Schlüer, H. Ney: Inverted Alignments for End-to-End Automatic Speech Recognition. IEEE Journal of selected topics in Signal Processing, Vol. 11, No. 8, pp. 1265-1273, Dec. 2017.

[Duda & Hart 73] R. O. Duda, P. E. Hart: Pattern Classification and Scene Analysis. Wiley, Hoboken, 1973.

[Forcada & Carrasco 05] M. L. Forcada, R. C. Carrasco: Learning the initial state of a second-order recurrent neural network during regular language inference. Neural Computation, Vol. 7, No. 5, pp. 923-930, Sep. 2005.

[Fontaine & Ris+ 97] V. Fontaine, C. Ris, J.-M. Boite: Nonlinear discriminant analysis for improved speech recognition. Eurospeech, Rhodes, Greece, Sep. 1997.

[Fritsch & Finke+ 97] J. Fritsch, M. Finke, A. Waibel: Adaptively Growing Hierarchical Mixtures of Experts. NIPS, Advances in Neural Information Processing Systems 9, MIT Press, pp. 459-465, 1997.

[Gemello & Manai+ 06] R. Gemello, F. Mana, S. Scanzio, P. Lafac, R. De Mori: Adaptation of Hybrid ANN/HMM Models Using Linear Hidden Transformations and Conservative Training. IEEE Int. Conf. on Acoustics Speech and Signal Processing Proceedings, Toulouse, 2006.

[Gers & Schmidhuber+ 00] F. A. Gers, J. Schmidhuber, F. Cummin: Learning to forget: Continual prediction with LSTM. Neural computation, Vol 12, No. 10, pp. 2451-2471, 2000.

[Gers & Schraudolph+ 02] F. A. Gers, N. N. Schraudolph, J. Schmidhuber: Learning precise timing with LSTM recurrent networks. Journal of Machine Learning Research, Vol. 3, pp. 115-143, 2002.

[Graves 12] A. Graves: Sequence Transduction with Recurrent Neural Networks. U of Toronto, Canada, arxiv, 12-Nov-2012.

[Graves & Fernandez+ 06] A. Graves, S. Fernandez, F Gomez, J. Schmidhuber: Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. Int. Conf. on Machine Learning, Pittsburgh, PA, pp. 369-376, 2006.

[Graves & Schmidhuber 09] A. Graves, J. Schmidhuber: Offline handwriting recognition with multidimensional recurrent neural networks. NIPS 2009.

[Grezl & Fousek 08] F. Grezl, P. Fousek: Optimizing bottle-neck features for LVCSR. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 4729-4732, Las Vegas, NV, March 2008.

[Grosicki & El Abed 09] E. Grosicki, H. El Abed: ICDAR 2009 Handwriting Recognition Competition. Int. Conf. on Document Analysis and Recognition (ICDAR) 2009, Barcelona, pp. 139-1402, July 2009.

[Haffner 93] P. Haffner: Connectionist Speech Recognition with a Global MMI Algorithm. 3rd Europ. Conf. on Speech Communication and Technology (Eurospeech'93), Berlin, Germany, Sep. 1993.

[Heigold & Macherey 05+] G. Heigold, W. Macherey, R. Schlüter, H. Ney: Minimum Exact Word Error Training. IEEE ASRU workshop, pp. 186-190, San Juan, Puerto Rico, Nov. 2005.

[Heigold & Schlüter 12+] G. Heigold, R. Schlüter, H. Ney, S. Wiesler: Discriminative Training for Automatic Speech Recognition: Modeling, Criteria, Optimization, Implementation, and Performance. IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 58-69, Nov. 2012.

[Hermansky & Ellis+ 00] H. Hermansky, D. W. Ellis, S. Sharma: Tandem connectionist feature extraction for conventional HMM systems. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 1635-1638, Istanbul, Turkey, June 2000.

[Hinton & Osindero+ 06] G. E. Hinton, S. Osindero, Y. Teh: A fast learning algorithm for deep belief nets. Neural Computation, Vol. 18, No. 7, pp. 1527-1554, July 2006.

[Hochreiter & Schmidhuber 97] S. Hochreiter, J. Schmidhuber: Long short-term memory. Neural Computation, Vol. 9, No. 8, pp. 1735–1780, Nov. 1997.

[Ivakhnenko 71] A. G. .Ivakhnenko: Polynomial theory of complex systems. IEEE Transactions on Systems, Man and Cybernetics, Vol. 1, No. 4, pp. 364-378, Oct. 1971.

[Jelinek & Mercer+ 77] F. Jelinek, R. L. Mercer, L. R. Bahl: Perplexity – a measure of the difficulty of speech recognition tasks. Journal of the Acoustical Society of America, 1977.

[Kaltenbrenner & Blunsom 13] N. Kalchbrenner, P. Blunsom: Recurrent continuous translation models. EMNLP 2013.

[Klakow & Peters 02] D. Klakow, J. Peters: Testing the correlation of word error rate and perplexity. Speech Communication, pp. 19–28, 2002.

[Koehn & Och+ 03] P. Koehn, F. J. Och, D. Marcu: Statistical Phrase-Based Translation. HLT-NAACL 2003, pp. 48-54, Edmonton, Canada, May-June 2003.

[Le & Allauzen+ 12] H.S. Le, A. Allauzen, F. Yvon: Continuous space translation models with neural networks. NAACL-HLT 2012, pp. 39-48, Montreal, QC, Canada, June 2002.

[LeCun & Bengio^{+} 94] Y. LeCun, Y. Bengio: Word-level training of a handwritten word recognizer based on convolutional neural networks. Int. Conf. on Pattern Recognition, Jerusalem, Israel, pp. 88-92, Oct. 1994.

[Makhoul & Schwartz 94] J. Makhoul, R Schwartz: State of the Art in Continuous Speech Recognition. Chapter 14, pp. 165-198, in D. B. Roe, J. G. Wilpon (Editors): Voice Communication Between Humans and Machines. National Academy of Sciences, 1994.

[Miao & Metze 15] Y. Miao. F Metze: On speaker adaptation of long short-term memory recurrent neural networks. Interspeech, Dresden, Germany, 2015.

[Mikolov & Corrado^{+} 13] T. Mikolov, G. Corrado, K. Chen, J. Dean: Efficient Estimation of Word Representations in Vector Space. Google, arxiv, 07-Sep-2013.

[Mikolov & Karafiat^{+} 10] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, S. Khudanpur: Recurrent neural network based language model. Interspeech, pp. 1045-1048, Makuhari, Chiba, Japan, Sep. 2010.

[Mohamed & Dahl^{+} 09] A. Mohamed, G. Dahl, G. Hinton: Deep belief networks for phone recognition. NIPS Workshop Deep Learning for Speech Recognition and Related Applications, 2009.

[Morgan & Bourlard 90] N. Morgan, H. Bourlard: Continuous speech recognition using multilayer perceptrons with hidden Markov models. ICASSP 1990, pp. 413-416, Albuquerque, NM, 1990.

[Nakamura & Shikano 89] M. Nakamura, K. Shikano: A Study of English Word Category Prediction Based on Neural Networks. ICASSP 89, p. 731-734, Glasgow, UK, May 1989.

[Neco & Forcada 97] R. P. Neco, M. L. Forcada: Asynchronous translations with recurrent neural nets. IEEE Int. Conf. on Neural Networks, pp. 2535-2540, June 1997.

[Ney 03] H. Ney: On the Relationship between Classification Error Bounds and Training Criteria in Statistical Pattern Recognition. First Iberian Conf. on Pattern Recognition and Image Analysis, Puerto de Andratx, Spain, Springer LNCS Vol. 2652, pp. 636-645, June 2003.

[Ney 84] H. Ney: The Use of a One–Stage Dynamic Programming Algorithm for Connected Word Recognition. IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-32, No. 2, pp. 263-271, April 1984.

[Ney & Haeb-Umbach+ 92] H. Ney, R. Haeb-Umbach, B.-H. Tran, M. Oerder: Improvements in Beam Search for 10000-Word Continuous Speech Recognition. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, San Francisco, CA, pp. 13-16, March 1992.

[Normandin & Cardin+ 94] Y. Normandin, R. Cardin, R. De Mori: High-Performance Connected Digit Recognition Using Maximum Mutual Information Estimation. IEEE Trans. on Speech and Audio Processing, vol. 2, no. 2, pp. 299-311, April 1994.

[Ouyang & Wu+ 22] L. Ouyang, J. Wu, X. Jiang et al.: Training language models to follow instructions with human feedback. OpenAI, arxiv, 04-Mar-2022.

[Och & Ney 03] F. J. Och, H. Ney: A Systematic Comparison of Various Alignment Models. *Computational Linguistics,* Vol. 29, No. 1, pp. 19-51, March 2003.

[Och & Ney 04] F. J. Och, H. Ney: The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, Vol. 30, No. 4, pp. 417-449, Dec. 2004.

[Och & Tillmann+ 99] F. J. Och, C. Tillmann, H. Ney: Improved Alignment Models for Statistical Machine Translation. Joint ACL/SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora, College Park, MD, pp. 20-28, June 1999.

[Patterson & Womack 66] J. D. Patterson, B. F. Womack: An Adaptive Pattern Classification Scheme. IEEE Trans. on Systems, Science and Cybernetics, Vol.SSC-2, pp.62-67, Aug. 1966.

[Povey & Woodland 02] D. Povey, P.C. Woodland: Minimum phone error and I-smoothing for improved discriminative training. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 105–108, Orlando, FL, May 2002.

[Printz & Olsen 02] H. Printz, P. A. Olsen: Theory and practice of acoustic confusability. Computer Speech and Language, pp. 131–164, Jan. 2002.

[Radford & Wu+ 18] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever: Language Models are Unsupervised Multitask Learners. OpenAI (GPT-2), preprint, 2018.

[Radford & Narasimhan+ 19] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever: Improving Language Understanding by Generative Pre-Training. OpenAI (GPT-1), preprint, 2019.

[Raissi & Beck+ 20] T. Raissi, E. Beck, R. Schlüter, H. Ney: Context-Dependent Acoustic Modeling without Explicit Phone Clustering arxiv, 2020.

[Raissi & Beck+ 21] T. Raissi, E. Beck, R. Schlüter, H. Ney: Towards Consistent Hybrid HMM Acoustic Modeling. arxiv, 2021.

[Raissi & Beck+ 22] T. Raissi, E. Beck, R. Schlüter, H. Ney: Improving Factored Hybrid HMM Acoustic Modeling without State Tying. arxiv, 2022.

[Robinson 94] A. J. Robinson: An Application of Recurrent Nets to Phone Probability Estimation. IEEE Trans. on Neural Networks, Vol. 5, No. 2, pp. 298-305, March 1994.

[Sainath & Weiss+ 16] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani: Factored Spatial and Spectral Multichannel Raw Waveform CLDNNs, Proc. ICASSP, 2016.

[Saon & Tüske+ 2021] G. Saon, Z. Tüske, D. Bolanos, B. Kingsbury: Advancing RNN Transducer Technology for Speech Recognition. IBM Research AI, Yorktown Heights, USA, arxiv, 17-Mar-2021.

[Sakoe & Chiba 71] H. Sakoe, S. Chiba: A Dynamic Programming Approach to Continuous Speech Recognition. Proc. 7th Int. Congr. on Acoustics, Budapest, Hungary, Paper 20 C 13, pp. 65-68, August 1971.

[Sak & Shannon+ 17] H. Sak, M. Shannon, K. Rao, F. Beaufays: Recurrent Neural Aligner: An Encoder-Decoder Neural Network Model for Sequence to Sequence Mapping. Interspeech, Stockholm, Sweden, pp. 1298-1302, Aug. 2017.

[Schlüter & Beck+ 19] R. Schlẗer, E. Beck, H. Ney: Upper and Lower Tight Error Bounds for Feature Omission with an Extension to Context Reduction. IEEE Trans. Pattern Anal. Mach. Intell., Vol. 41, No. 2, pp. 502-514, 2019.

[Schlüter & Nussbaum+ 11] R. Schlüter, M. Nussbaum-Thom, H. Ney: On the Relationship between Bayes Risk and Word Error Rate in ASR. IEEE Trans. on Audio, Speech, and Language Processing, vol. 19, no. 5, p. 1103-1112, July 2011.

[Schlüter & Nussbaum+ 12] R. Schlüter, M. Nussbaum-Thom, H. Ney: Does the Cost Function Matter in Bayes Decision Rule? IEEE Trans. PAMI, No. 2, pp. 292–301, Feb. 2012.

[Schlüter & Nussbaum-Thom+ 13] R. Schlüter, M. Nußbaum-Thom, E. Beck, T. Alkhouli, H. Ney: Novel Tight Classification Error Bounds under Mismatch Conditions based on f-Divergence. IEEE Information Theory Workshop, pp. 432–436, Sevilla, Spain, Sep. 2013.

[Schlüter & Scharrenbach+ 05] R. Schlüter, T. Scharrenbach, V. Steinbiss, H. Ney: Bayes Risk Minimization using Metric Loss Functions Interspeech, pages 1449-1452, Lisboa, Portugal, Sep. 2005.

[Schuster & Paliwal 97] M. Schuster, K. K. Paliwal: Bidirectional Recurrent Neural Networks. IEEE Trans. on Signal Processing, Vol. 45, No. 11, pp. 2673-2681, Nov. 1997.

[Schwenk 07] H. Schwenk: Continuous space language models. Computer Speech and Language, Vol. 21, No. 3, pp. 492–518, July 2007.

[Schwenk 12] H. Schwenk: Continuous Space Translation Models for Phrase-Based Statistical Machine Translation. 24th Int. Conf. on Computational Linguistics (COLING), Mumbai, India, pp. 1071–1080, Dec. 2012.

[Schwenk & Costa-jussa+ 07] H. Schwenk , M. R. Costa-jussa, J. A. R. Fonollosa: Smooth bilingual n-gram translation. Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 430–438, Prague, June 2007.

[Schwenk & Déchelotte+ 06] H. Schwenk, D. Déchelotte, J. L. Gauvain: Continuous Space Language Models for Statistical Machine Translation. COLING/ACL 2006, pp. 723–730, Sydney, Australia July 2006.

[Schwenk & Gauvain 02] H. Schwenk, J.-L. Gauvain: Connectionist language modeling for large vocabulary continuous speech recognition. pp. 765-768, ICASSP 2002.

[Seide & Li+ 11] F. Seide, G. Li, D. Yu: Conversational Speech Transcription Using Context-Dependent Deep Neural Networks. Interspeech, pp. 437-440, Florence, Italy, Aug. 2011.

[Solla & Levin+ 88] S. A. Solla, E. Levin, M. Fleisher: Accelerated Learning in Layered Neural Networks. Complex Systems, Vol.2, pp. 625-639, 1988.

[Soltan & Ananthakrishnan+ 22] S. Soltan, S. Ananthakrishnan, J. FitzGerald et al.: AlexaTM 20B: Few-Shot Learning Using a Large-Scale Multilingual Seq2seq Model. Amazon, arxiv, 03-Aug-2022.

[Stolcke & Grezl+ 06] A. Stolcke, F. Grezl, M.-Y. Hwang, X. Lei, N. Morgan, D. Vergyri: Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Toulouse, France, May 2006.

[Sundermeyer & Alkhouli+ 14] M. Sundermeyer, T. Alkhouli, J. Wuebker, H. Ney: Translation Modeling with Bidirectional Recurrent Neural Networks. Conf. on Empirical Methods in Natural Language Processing (EMNLP), pp. 14–25, Doha, Qatar, Oct. 2014.

[Sundermeyer & Ney+ 15] M. Sundermeyer, H. Ney, R. Schlüter: From feedforward to recurrent LSTM neural networks for language modeling. IEEE/ACM Trans. on Audio, Speech, and Language Processing, Vol. 23, No. 3, pp. 13–25, March 2015.

[Sundermeyer & Schlüter+ 12] M. Sundermeyer, R. Schlüter, H. Ney: LSTM neural networks for language modeling. Interspeech, pp. 194–197, Portland, OR, USA, Sep. 2012.

[Sutskever & Vinyals+ 14] I. Sutskever, O. Vinyals, Q. V. Le: Sequence to Sequence Learning with Neural Networks. Google, arxiv, 14-Dec-2014.

[Tüske & Plahl+ 11] Z. Tüske, C. Plahl, R. Schlüter: A study on speaker normalized MLP features in LVCSR. Interspeech, pp. 1089-1092, Florence, Italy, Aug. 2011.

[Tüske & Golik+ 14] Z. Tüske, P. Golik, R. Schlúter, H. Ney: Acoustic Modeling with Deep Neural Networks Using Raw Time Signal for LVCSR. Interspeech, ISCA best student paper award, pp. 890-894, Singapore, Sep. 2014.

[Utgoff & Stracuzzi 02] P. E. Utgoff, D. J. Stracuzzi: Many-layered learning. Neural Computation, Vol. 14, No. 10, pp. 2497-2539, Oct. 2002.

[Valente & Vepa+ 07] F. Valente, J. Vepa, C. Plahl, C. Gollan, H. Hermansky, R. Schlüter: Hierarchical Neural Networks Feature Extraction for LVCSR system. Interspeech, pp. 42-45, Antwerp, Belgium, Aug. 2007.

[Vapnik 98] Vapnik: Statistical Learning Theory. Addison-Wesley, 1998.

[Variani & Sainath+ 16] E. Variani, T. N. Sainath, I. Shafran, M. Bacchiani: Complex Linear Projection (CLP): A Discriminative Approach to Joint Feature Extraction and Acoustic Modeling. Interspeech 2016, San Francisco, CA, pp. 808-812, Sep. 2016.

[Vaswani & Shazeer+ 17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser: Attention Is All You Need. Google, arxiv, 06-Dec-2017.

[Vaswani & Zhao+ 13] A. Vaswani, Y. Zhao, V. Fossum, D. Chiang: Decoding with Large-Scale Neural Language Models Improves Translation. Conf. on Empirical Methods in Natural Language Processing (EMNLP, pp. 1387–1392, Seattle, Washington, USA, Oct. 2013.

[Velichko & Zagoruyko 70] V. M. Velichko, N. G. Zagoruyko: Automatic Recognition of 200 Words. Int. Journal Man-Machine Studies, Vol. 2, pp. 223-234, June 1970.

[Vintsyuk 68] T. K. Vintsyuk: Speech Discrimination by Dynamic Programming. Kibernetika (Cybernetics), Vol. 4, No. 1, pp. 81-88, Jan.-Feb. 1968.

[Vintsyuk 71] T. K. Vintsyuk: Elementwise Recognition of Continuous Speech Composed of Words from a Specified Dictionary. Kibernetika (Cybernetics), Vol. 7, pp. 133-143, March-April 1971.

[Vogel & Ney+ 96] S. Vogel, H. Ney, C. Tillmann: HMM-based word alignment in statistical translation. Int. Conf. on Computational Linguistics (COLING), pp. 836-841, Copenhagen, Denmark, Aug. 1996.

[Waibel & Hanazawa+ 88] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. L. Lang: Phoneme Recognition: Neural Networks vs. Hidden Markov Models. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), New York, NY, pp.107-110, April 1988.

[Wang & Alkhouli+ 17] W. Wang, T. Alkhouli, D. Zhu, H. Ney: Hybrid Neural Network Alignment and Lexicon Model in Direct HMM for Statistical Machine Translation. Annual Meeting ACL, pp. 125-131, Vancouver, Canada, Aug. 2017.

[Wang & Zhu+ 18] W. Wang, D. Zhu, T. Alkhouli, Z. Gan, H. Ney: Neural Hidden Markov Model for Machine Translation. Annual Meeting ACL, Melbourne, Australia, July 2018.

[Xu & Povey+ 10] H. Xu, D. Povey, L. Mangu, J. Zhu: Minimum Bayes Risk Decoding and System Combination Based on a Recursion for Edit Distance. Computer Speech and Language, Sep. 2010.

[Zens & Och+ 02] R. Zens, F. J. Och, H. Ney: Phrase-Based Statistical Machine Translation. 25th Annual German Conf. on AI, pp. 18–32, LNAI, Springer 2002.

[Zhou & Berger+ 2021] W. Zhou, S. Berger, R. Schlüter, H. Ney: Phoneme Based Neural Transducer for Large Vocabulary Speech Recognition. ICASSP, Toronto, June 2021.

[Zhou & Zeyer+ 2021] W. Zhou, A. Zeyer, A. Merboldt, R .Schlüter, H. Ney: Equivalence of Segmental and Neural Transducer Modeling: A Proof of Concept. Interspeech, pp. 2891-2895, Graz, 2021.

END

RTTH, Jaca 2023: Data-Driven Speech & Language Technology:
From Small to Large Models