Sequence Learning & Speech Recognition

TU Kaiserslautern & DFKI

Image Understanding and Pattern Recognition

Prof. Dr. Thomas Breuel

Presentation by Martin Krämer

Contents

- Sequence Learning
- Hidden Markov Models
  - Basics, Example, Three Basic Problems, Trellis, Viterbi
- Speech Recognition
  - System Architecture, Hypothesis Search, A*-Algorithm, State-of-the-Art
- Speech Translation
  - System Architecture, Empirical Results
- Network Intrusion Detection
  - Multi-Staged Attacks, System Architecture, Empirical Results

References

- Gernot A. Fink: Mustererkennung mit Markov-Modellen, Teubner, 2003
- Frederick Jelinek: Statistical Methods for Speech Recognition, MIT Press, 1998
- Lawrence R. Rabiner: A Tutorial on HMMs and Selected Applications in Speech Recognition, Proceedings of the IEEE, Volume 77/2, p. 257-286, 1989
- Mukund Padmanabhan & Michael Picheny: Large-Vocabulary Speech Recognition Algorithms, IEEE Computer Journal, Volume 35/4, p. 42-50, 2002
- Dirk Ourston, Sara Matzner & William Stump: Applications of HMMs to Detecting Multi-Stage Network Attacks, Proceedings of the 36th Hawaii International Conference on System Sciences, Volume 5, 2003

Sequence Learning Overview

- analysis of a sequence of elements
- especially *time-discrete* systems

anomaly detection: recognition of deviant activity inside a stream of data (i.e. visual surveillance)

bioinformatics: sequence alignment, gene finding, protein-protein interaction, evolution modeling, ...

natural language processing: speech recognition, speech translation & speech understanding

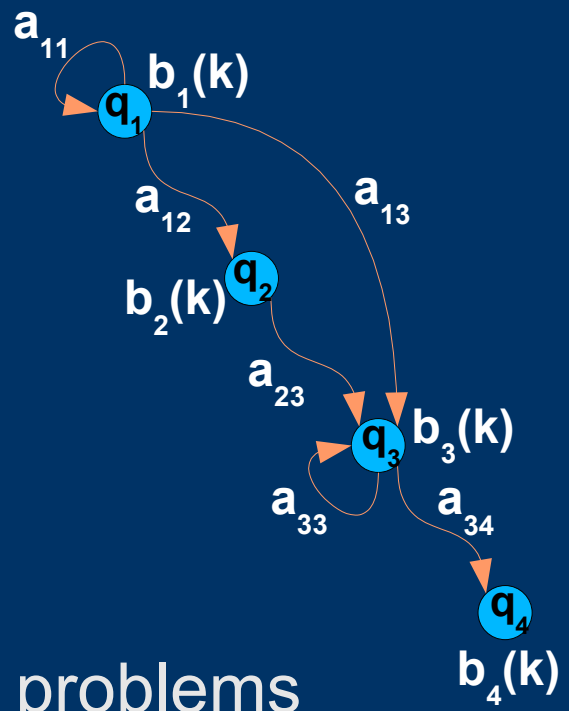
Hidden Markov Models Basics

- *two-stage* stochastic process

1st layer: discrete Markov process

2nd layer: generates emissions

- sequence of states is unknown
- only the emissions O_t are observable



- suitable for *time-oriented* processes
- extensively used for sequence learning problems

N hidden states
M observation symbols

HMM $\lambda = (A,B,\pi)$

$a_{ij} = P(q_{t+1}=j|q_t=i),\ 1 \le i,j \le N$

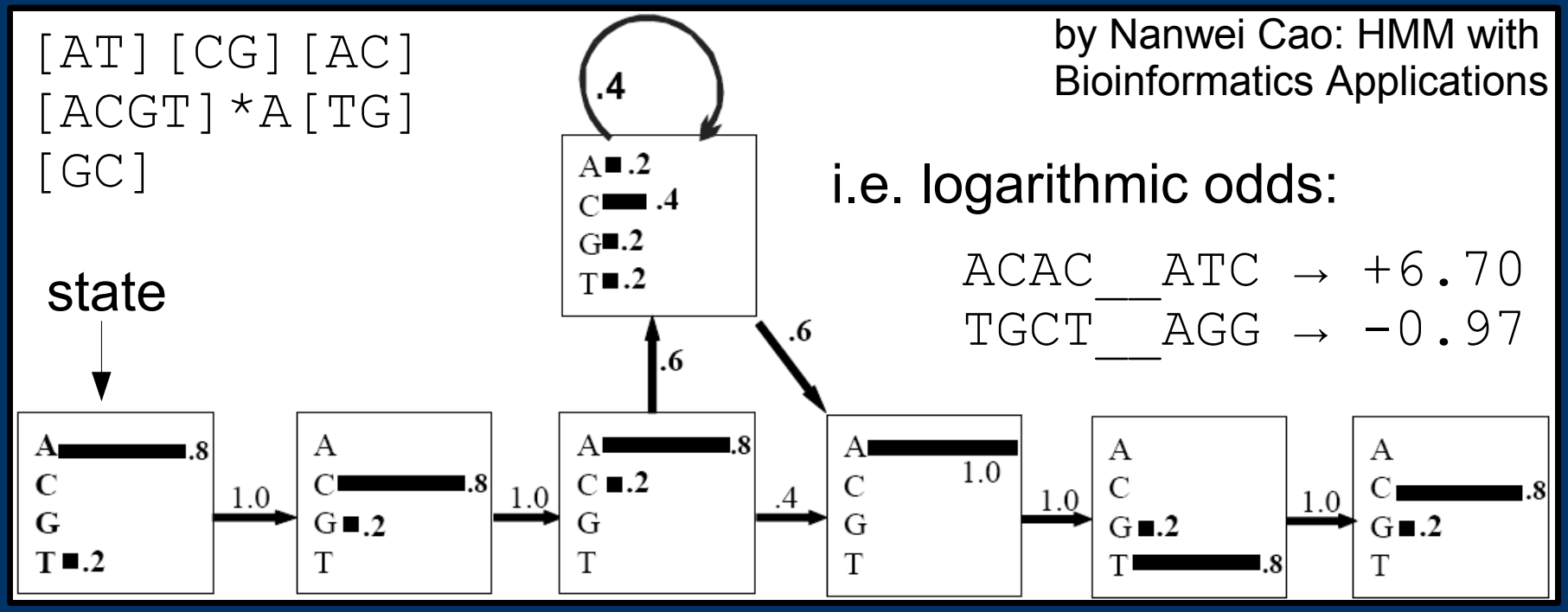
$b_j(k) = P(o_t=k|q_t=j),\ 1 \le k \le M, 1 \le j \le N$

$\pi_i = P(q_1=i),\ 1 \le i \le N$

Hidden Markov Models Example

- DNA matching:

```
ACA___ATG,TCAACTATC,ACAC__AGC,
AGA___ATC,ACCG__ATC
```



Hidden Markov Models
Three Basic Problems

Evaluation Problem: given observation sequence and model – how to efficiently compute the probability of the observation sequence?

→ *„Forward-Backward“-procedure*

Decoding Problem: given observation sequence and model – how to choose an optimal corresponding state sequence?

→ *„Viterbi“-algorithm*

Optimization Problem: how to adjust the model parameters to maximize the probability of the observation sequence?

→ *„Baum-Welch“-algorithm*

Hidden Markov Models
Basics of the Viterbi-Algorithm

given: model $\lambda = (A,B,\pi)$

given: observation sequence $O = O_1 O_2 ... O_n$

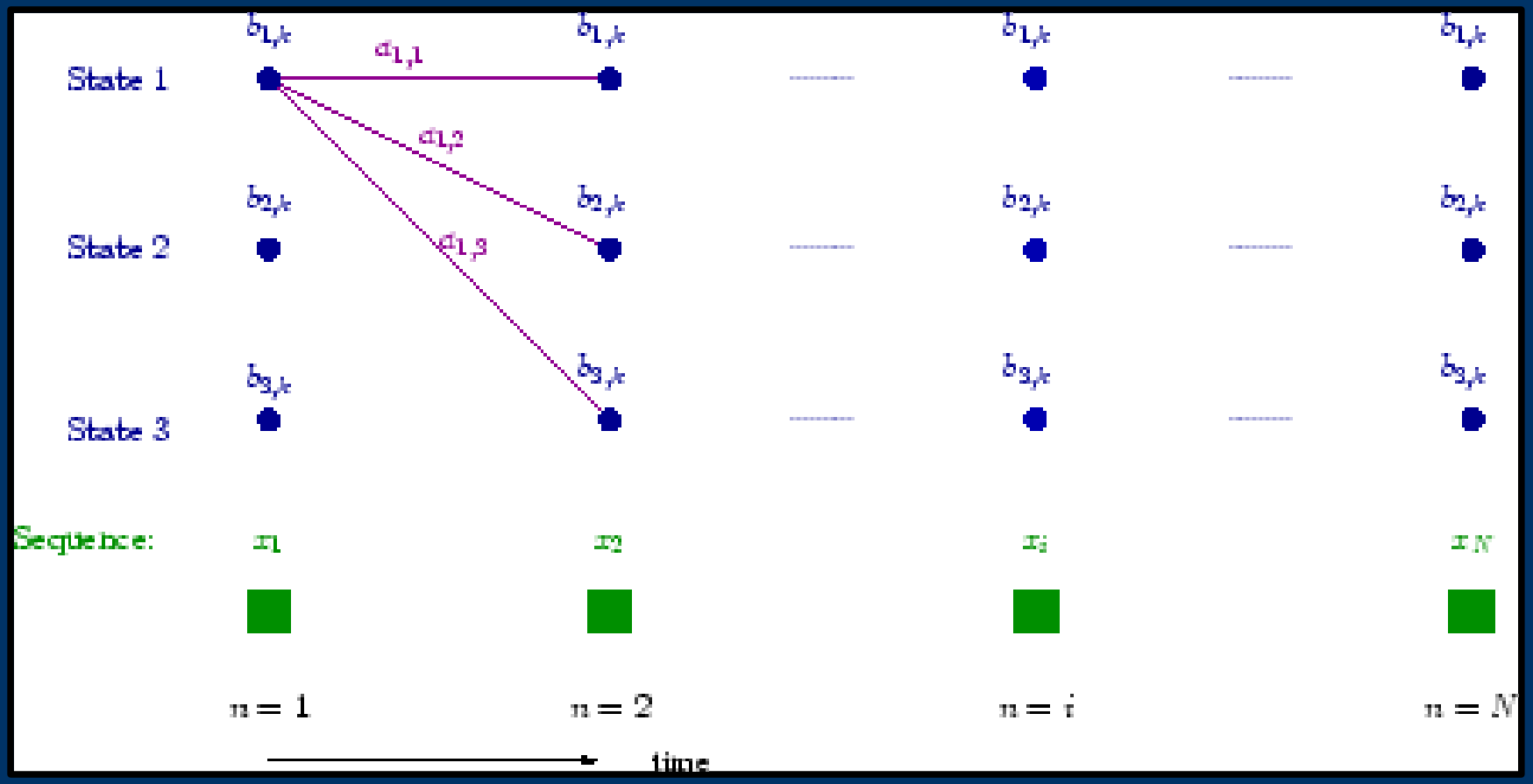
sought: state sequence $Q = Q_1 Q_2 ... Q_n$ (optimal)

- that means *maximizing $P(Q|O,\lambda)$*
- solution by dynamic programming

Viterbi algorithm: first traverse through a Trellis with backtracking; then after completion recurse back to the start determining the overall best path

Hidden Markov Models
Trellis Diagram

Example of a Trellis Diagram



Hidden Markov Models
Viterbi-Algorithm in Detail

we define: $\delta_n(i) = \max(Q_1 Q_2 ... Q_{n-1}) [P(Q_1 Q_2 ... Q_{n-1}, Q_n = i, O|\lambda)]$

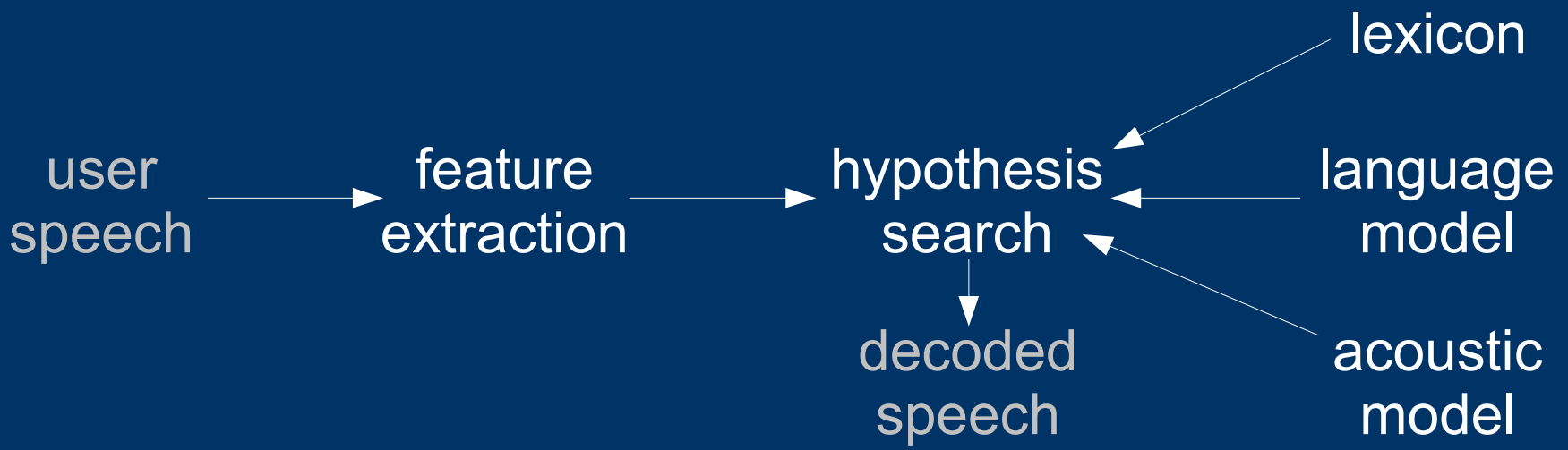
per induction: $\delta_{n+1}(j) = \max(i) [\delta_n(i) a_{ij}] b_j(O_{n+1})$

- **Initialization:** $\delta_1(i) = \pi_i \, b_i(O_1)$ *and* $\psi_1(i) = 0$ $(1 \le i \le N)$
- **Recursion:** $(2 \le n \le T$ *and* $1 \le j \le N)$
  - $\delta_n(j) = \max(1 \le i \le N) [\delta_{n-1}(i) a_{ij}] b_j(O_n)$
  - $\psi_n(j) = \arg\max(1 \le i \le N) [\delta_{n-1}(i) a_{ij}]$
- **Termination:**
  - $P^* = \max(1 \le i \le N) [\delta_T(i)]$
  - $Q_T^* = \arg\max(1 \le i \le N) [\delta_T(i)]$
- **Backtracking:** $Q_n^* = \psi_{n+1}(Q_{n+1}^*)$ where $n = T-1, T-2, ..., 1$

Speech Recognition System Architecture

- map sampled speech onto word sequence

ASR: *automatic speech recognition*, i.e. the capability (of computers) to understand naturally spoken language



Speech Recognition Hypothesis Search

- tries to determine the word sequence with the highest likelihood depending on speech input and model parameters

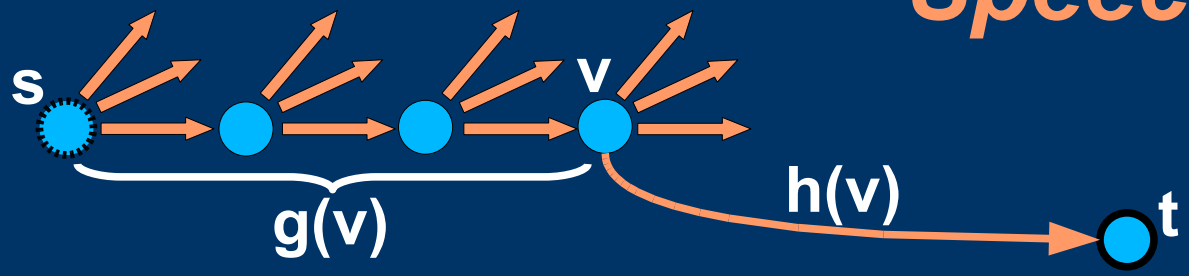
equivalent: search a tree whose branches are labelled with the words of a dictionary

➔ naive approach unfeasible because dictionaries are usually rather large

needed: heuristic evaluation function that generates a list of the best candidates

➔ some tree-search algorithm like A* can be used

Speech Recognition
A*-Algorithm



f(v) = g(v) + h(v)

- h(v): measured costs from v to target vertex t
- g(v): actual costs from start vertex s to v
- f(v): estimated costs from start s over v to target t

if h(v) does not overestimate, then A* is effective!

- first calculate all possible successors
- then heuristically evaluate each one of them
- finally follow the vertex v with minimal f(v)

Speech Recognition State-of-the-Art

quantity: *word error rate*

- improved by more training data, cleaner background, less complex language, less spontaneity, smaller vocabulary

usability: speech recognition feasible for task-dependant systems

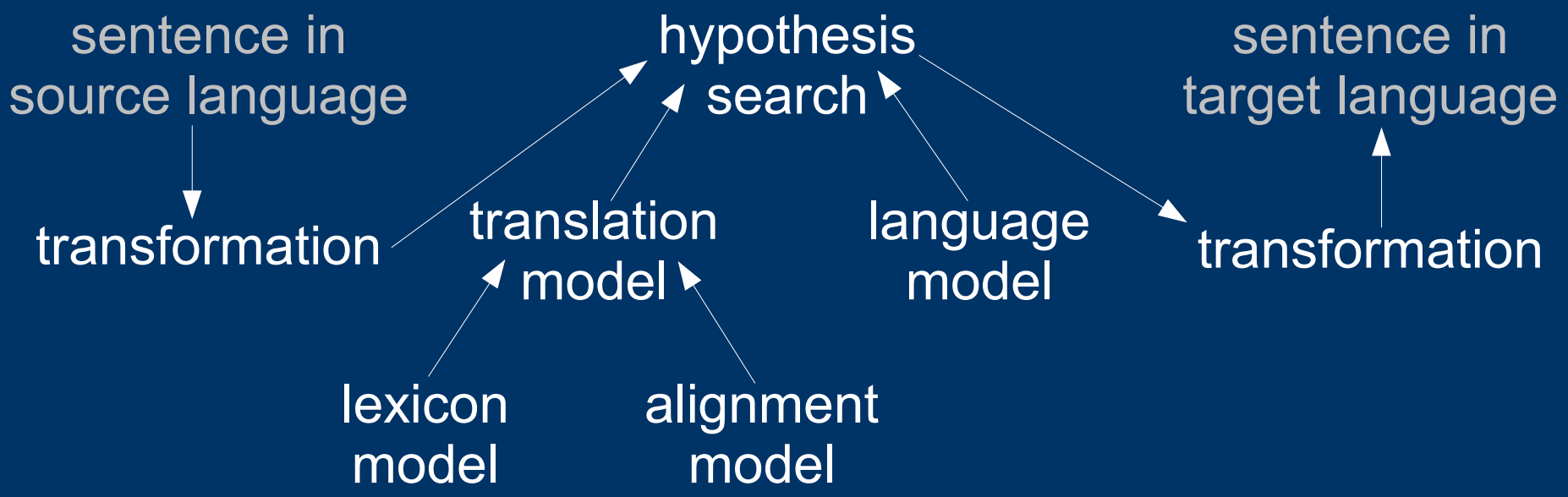
- ASR for conversational telephone speech yields 32.7% word error rate (human: approximately 5.0%)

problems: natural intonation, different speakers, background noise

Speech Translation System Architecture

- map sentences between different languages

general case of „Speech Understanding“ (natural to formal language used for semantic representation)



Speech Translation Empirical Results

VERBMOBIL: translate naturally spoken language between German and English, vocabulary: ~8000 words

- best results by statistical approach: 29% error rate
- worst results by rule-based approach: 62% error rate

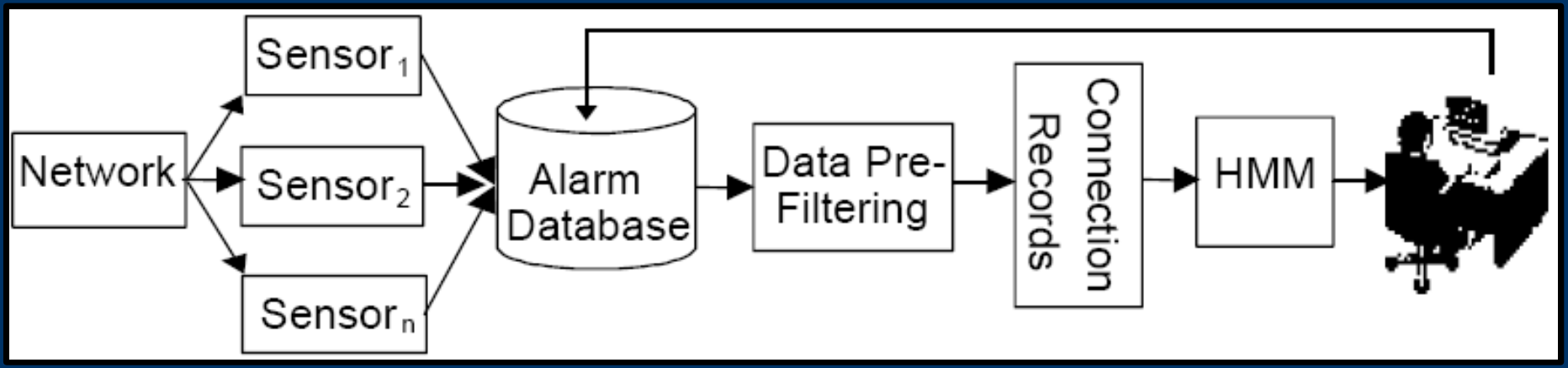
DARPA: translate between spoken/written and English/Chinese

- problem: mapping between small and large dictionaries
- best results by statistical approach
- so it is applicable for very different languages as well

Network Intrusion Detection Multi-Staged Attacks

- attacks consist of *several steps* that may occur over an extended period of time:
  - reconaissence
  - penetration
  - exploitation
  - consolidation
- attackers may perform steps to mask the intrusion
- action sequences may be altered due to lack of experience
- rule-based approaches may be able to identify individual stages of an attack more or less accurately
- machine learning techniques are needed to identify sophisticated attacks (i.e. HMMs, NNs, decision trees)

Network Intrusion Detection System Architecture

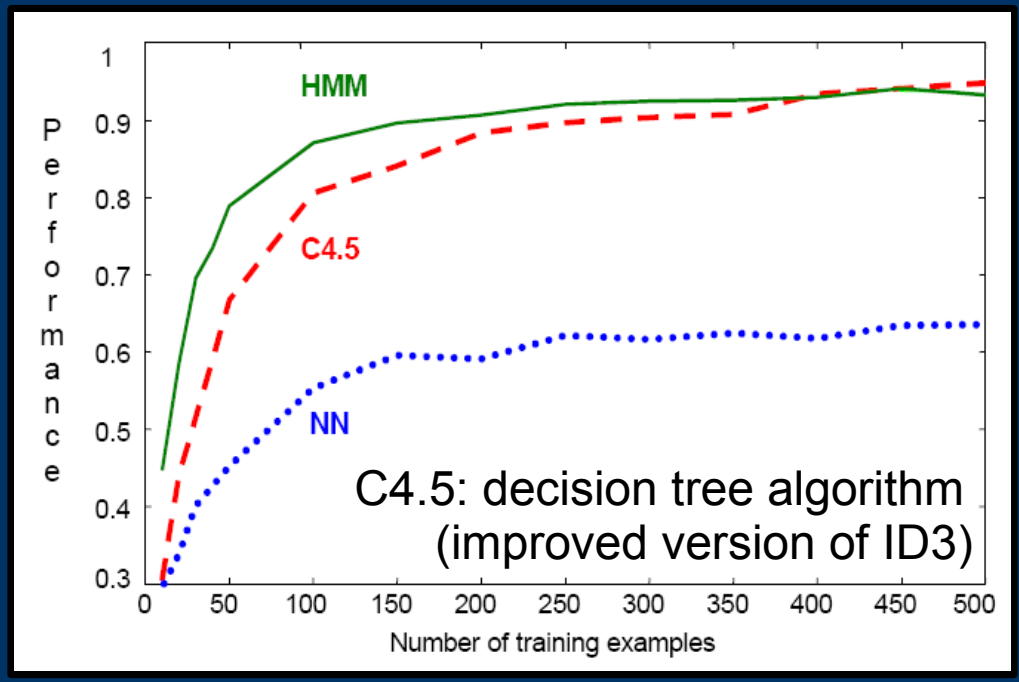


- sensors monitor traffic and set alerts on possible intrusions
- those alerts (observables) are stored in an alarm database
- redundant data is eliminated (repetition, false positives)
- connection records include source ip, target ip and an ordered sequence of alerts during a 24-hour period
- HMM determines the most probable type of attack; network analyst corrects false classifcations and updates the alarm database
- the hidden state sequence expresses the attacker's actions

Network Intrusion Detection Empirical Results

comparison of different machine learning techniques

- HMMs perform best
- C4.5 learns slower
- NNs are inferior to both



➔ the quick learning of HMMs indicates applicability to the „rare data“ problem (only few samples of attack type are available)

Sequence Learning & Speech Recognition

Thanks for your attention!

Questions?