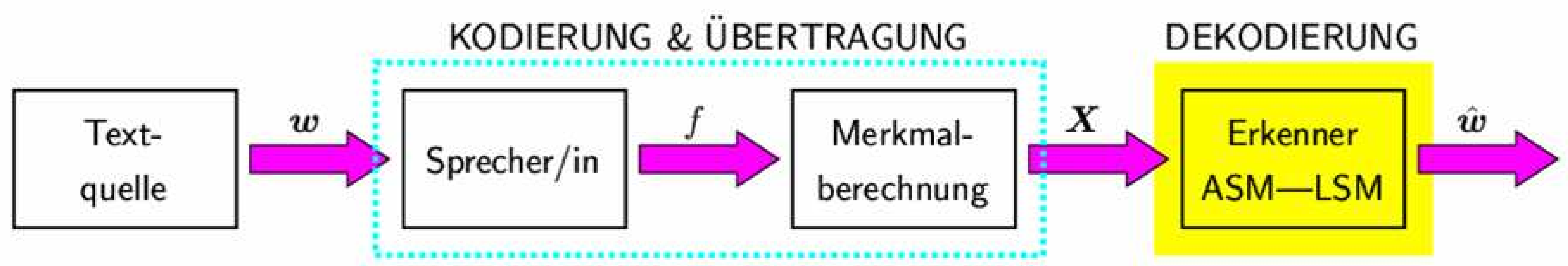
Wortdekodierung

Vorlesungsunterlagen
Speech Communication 2, SS 2004

Franz Pernkopf/Erhard Rank

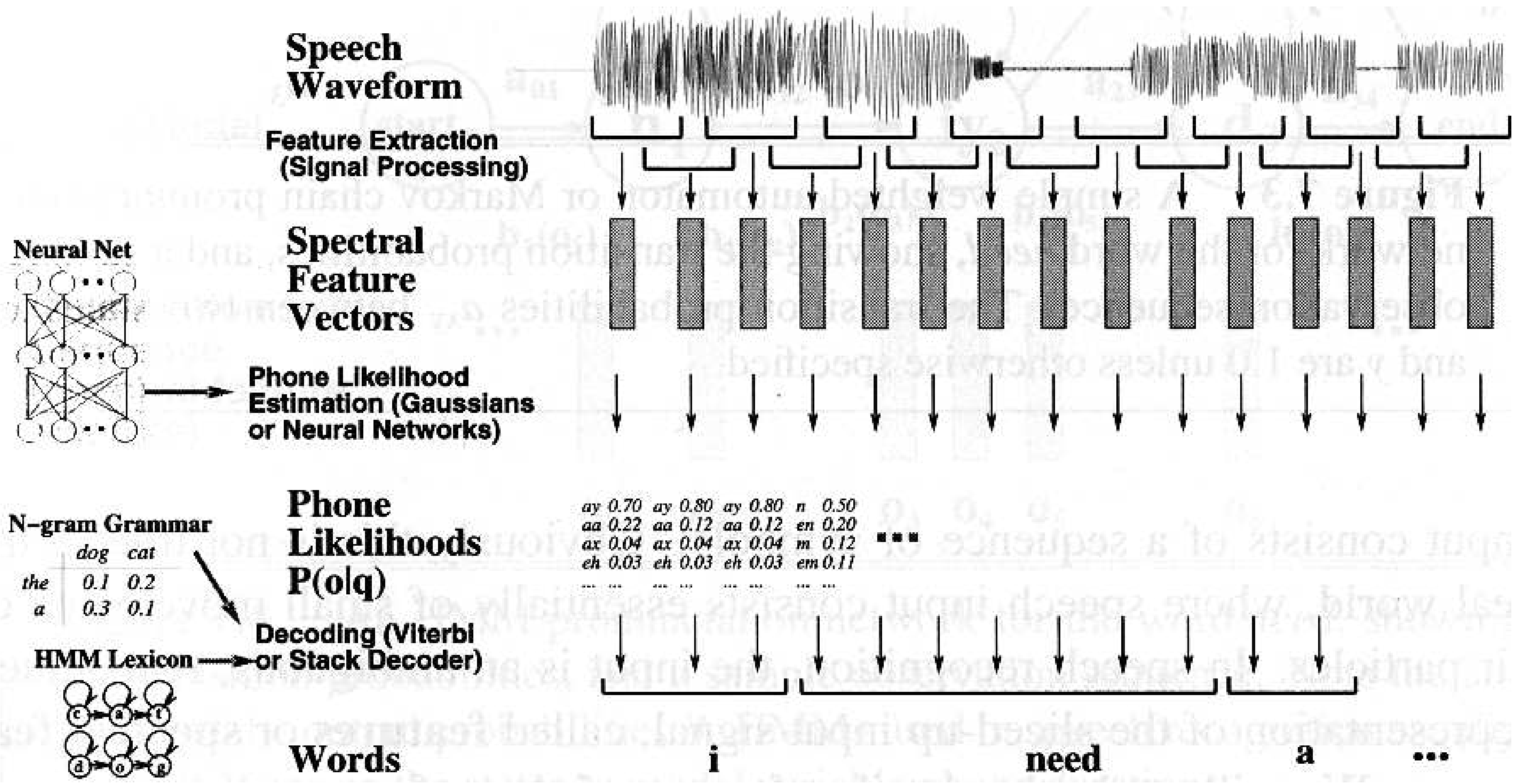
Institute of Signal Processing and Speech Communication
University of Technology Graz
Inffeldgasse 16c , 8010 Graz, Austria
Tel: +43 316 873 4436
E-Mail: pernkopf/rank@inw.tugraz.at

Kommunikationstheoretisches Modell der Spracherzeugung und -erkennung:



DEKODIERUNG = Maximierungsaufgabe

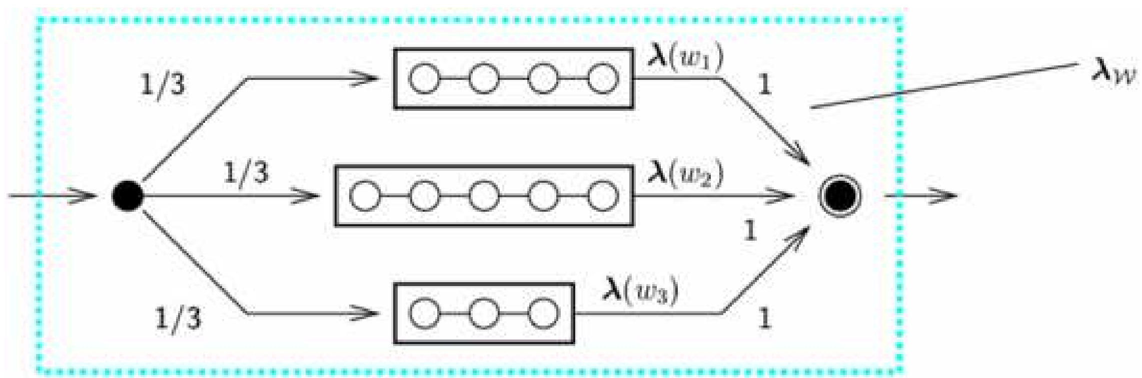
$$\hat{\boldsymbol{w}} = \operatorname*{argmax}_{\boldsymbol{v}} P(\boldsymbol{X} \mid \boldsymbol{v}) \cdot P(\boldsymbol{v})$$


Speech Waveform
Feature Extraction (Signal Processing)
Neural Net
Spectral Feature Vectors
Phone Likelihood Estimation (Gaussians or Neural Networks)
N-gram Grammar
dog cat
the 0.1 0.2
a 0.3 0.1
Phone Likelihoods P(o|q)
ay 0.70 aa 0.22 ax 0.04 eh 0.03
ay 0.80 aa 0.12 ax 0.04 eh 0.03
ay 0.80 aa 0.12 ax 0.04 eh 0.03
n 0.50 en 0.20 m 0.12 em 0.11
HMM Lexicon
Decoding (Viterbi or Stack Decoder)
Words
i
need
a

Kompilierte HMM-Netzwerke

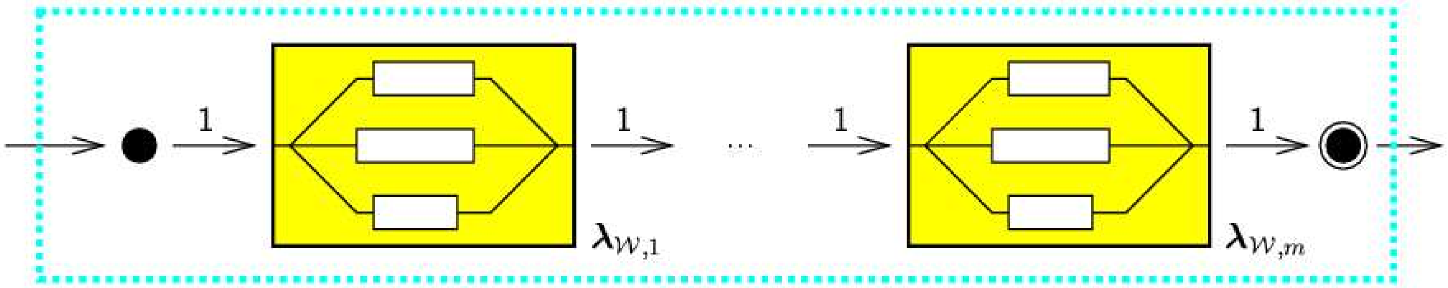
- Voraussetzung: einfache Wortübergangsgrammatik *(Bigramme)*
- Voraussetzung: jedes λ_w besitzt je einen A/E-Zustand
- Vernetzung der Wort-HMMs "im Sinne der Grammatik"
- Dekodierung: Viterbi-Algorithmus auf dem HMM-Netzwerk
- Optimale Zustandsfolge ⇒ Wahrscheinlichste Wortfolge

HMM-Netzwerk zur Einzelworterkennung:



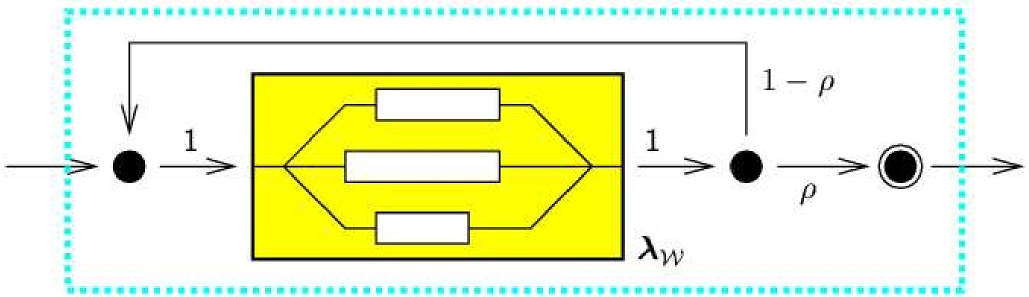
Verbundworterkennung

bei vorgegebener Satzlänge *m*



Verbundworterkennung

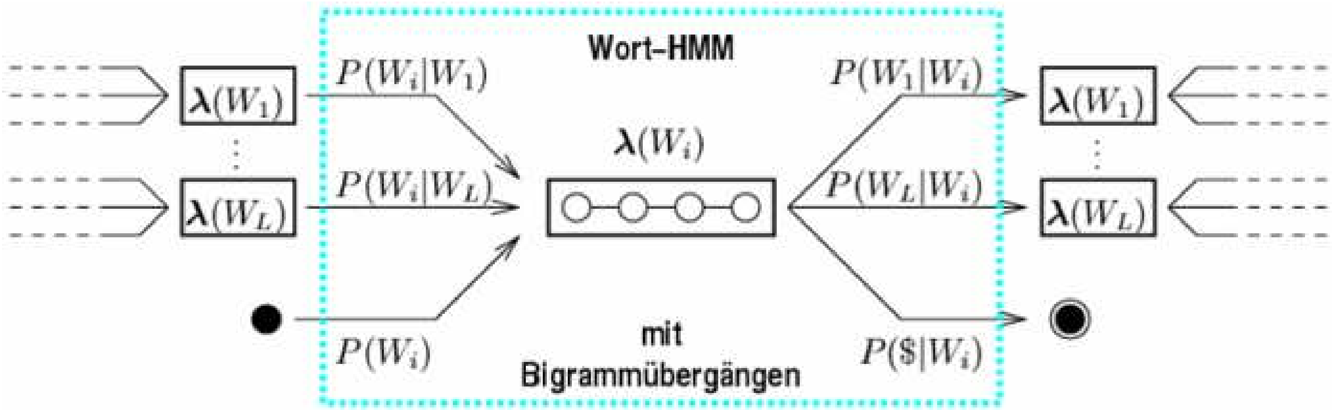
bei unbekannter Satzlänge



- ***Fluchtwahrscheinlichkeit*** ρ
  - Verlassern der Wiederholungschleife
- Vermeiden der “Kantenexplosion” durch ***konfluente*** Zustände

Verbundworterkennung

mit wortbezogener Bigrammgrammatik



- L Wortmodelle

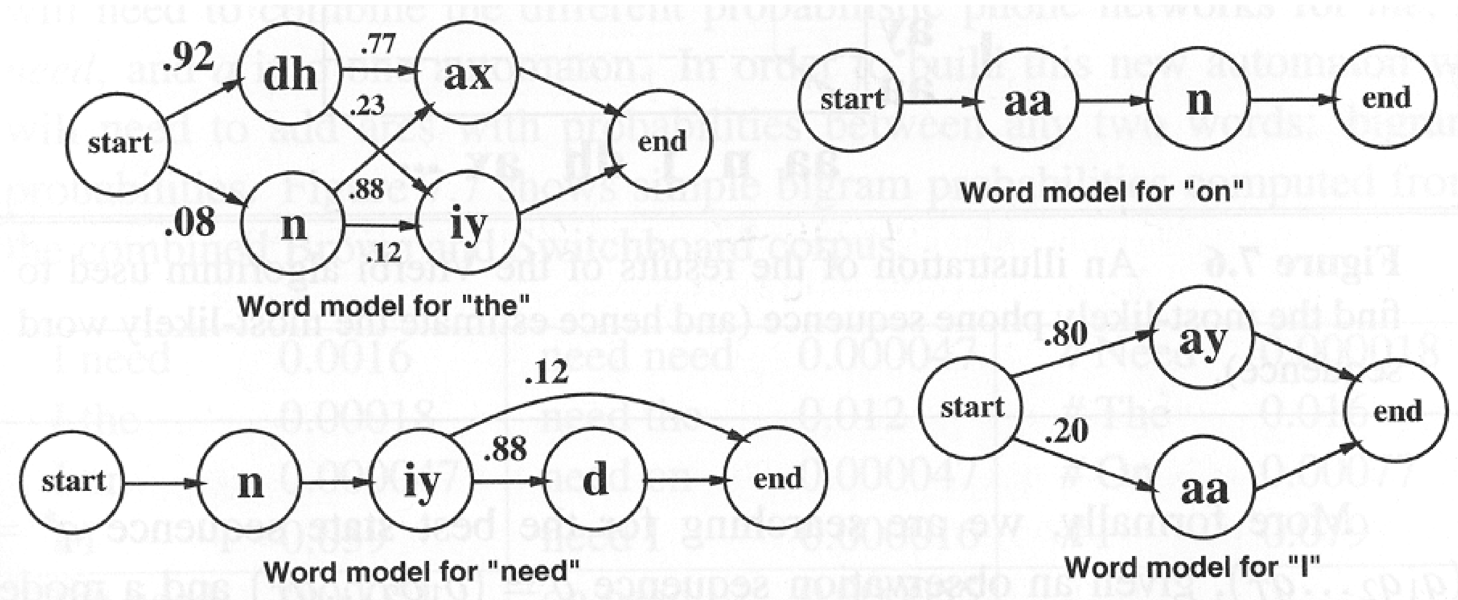
  (wie ein Einzel- oder grammatikfreier Verbundworterkenner)
- L^2 Wort-HMM-Übergangskanten mit Bigrammwahrscheinlichkeiten

Viterbi-Dekodierung

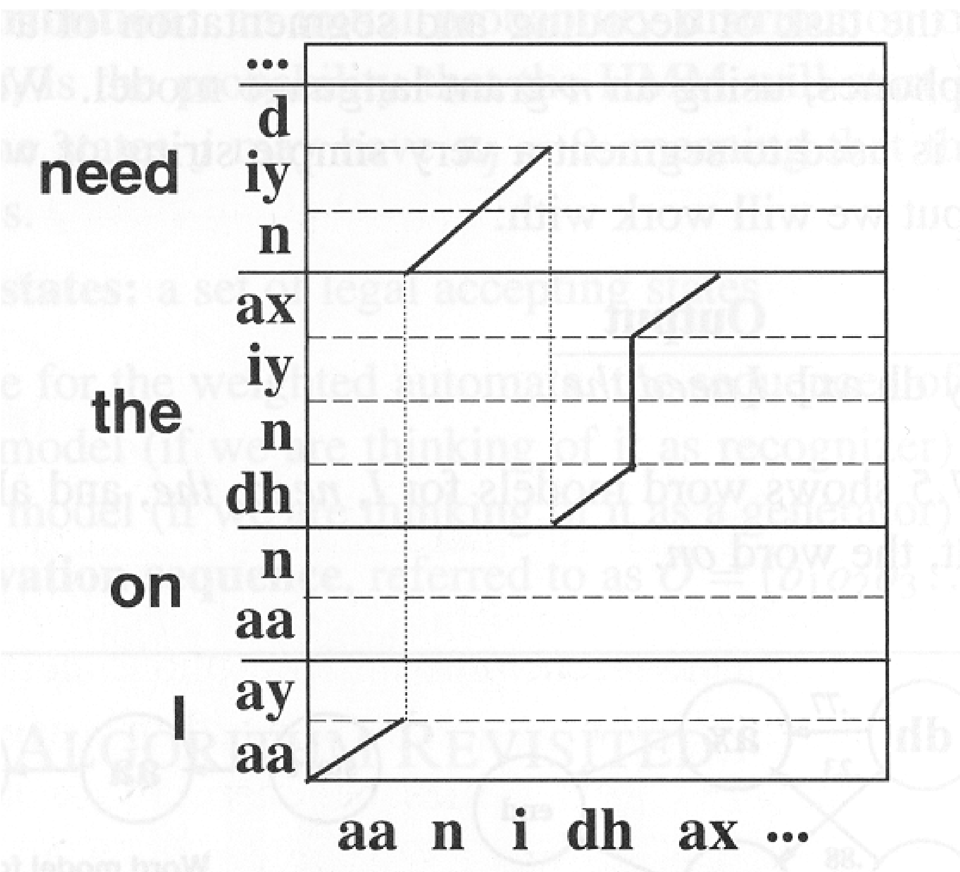
Example: Input: [aa n iy dh ax]

Output: *I need the*

Word models for *I*, *need*, *the*, and *on*:



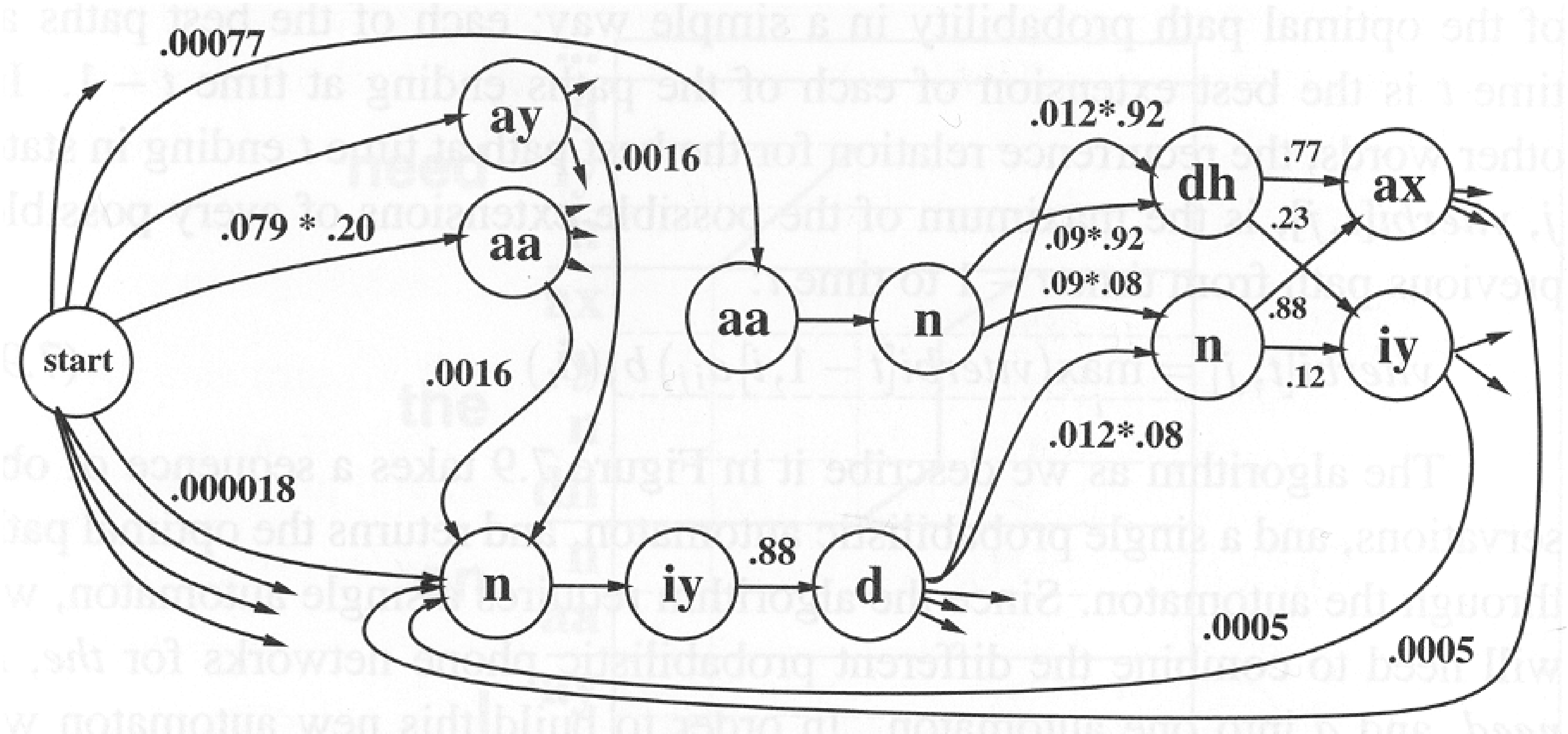
Viterbi algorithm used to find the most-likely phone sequence within word models:



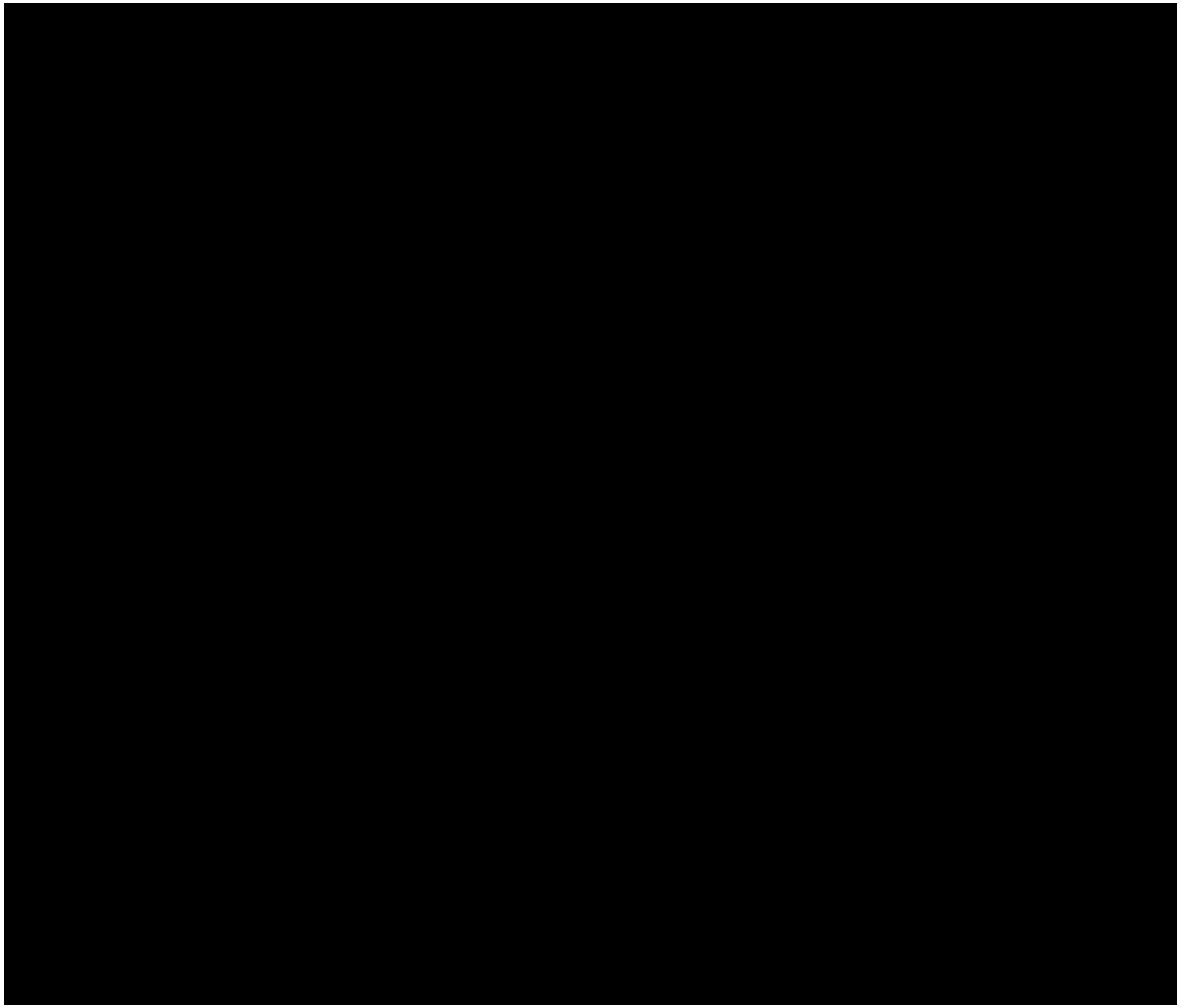
Example bigram probabilities for the words *the*, *on*, *need*, and *I* following each other, and starting a sentence (computed from Brown and Switchboard corpus).

I need	0.0016	*need need*	0.000047	*# Need*	0.000018
I the	0.00018	*need the*	0.012	*# The*	0.016
I on	0.000047	*need on*	0.000047	*# On*	0.00077
I I	0.039	*need I*	0.000016	*# I*	0.079
the need	0.00051	*on need*	0.000055		
the the	0.0099	*on the*	0.094		
the on	0.00022	*on on*	0.0031		
the I	0.00051	*on I*	0.00085		

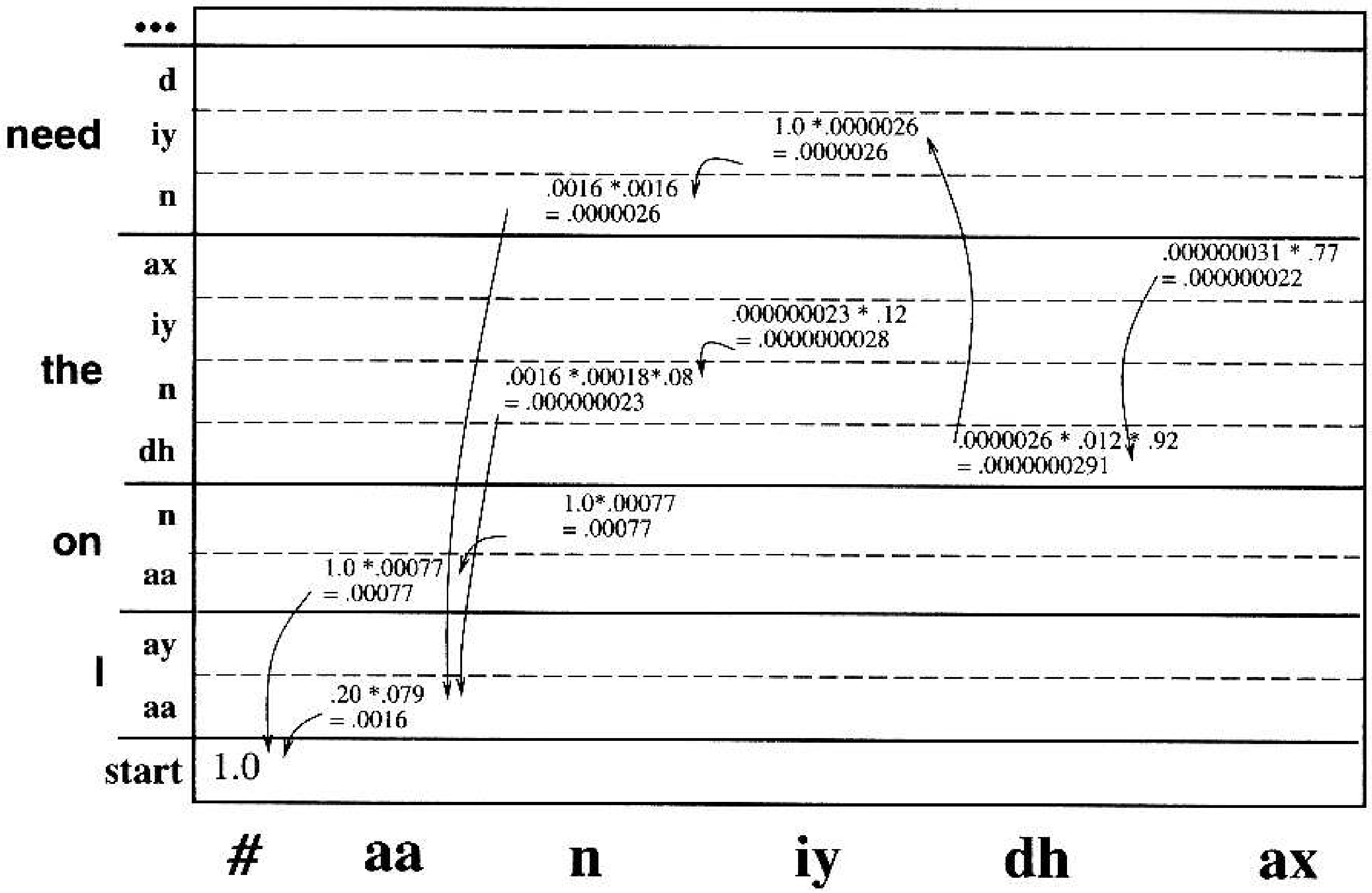
Finite state machine for the words *I*, *need*, *on*, and *the*:



Viterbi-Dekodierung

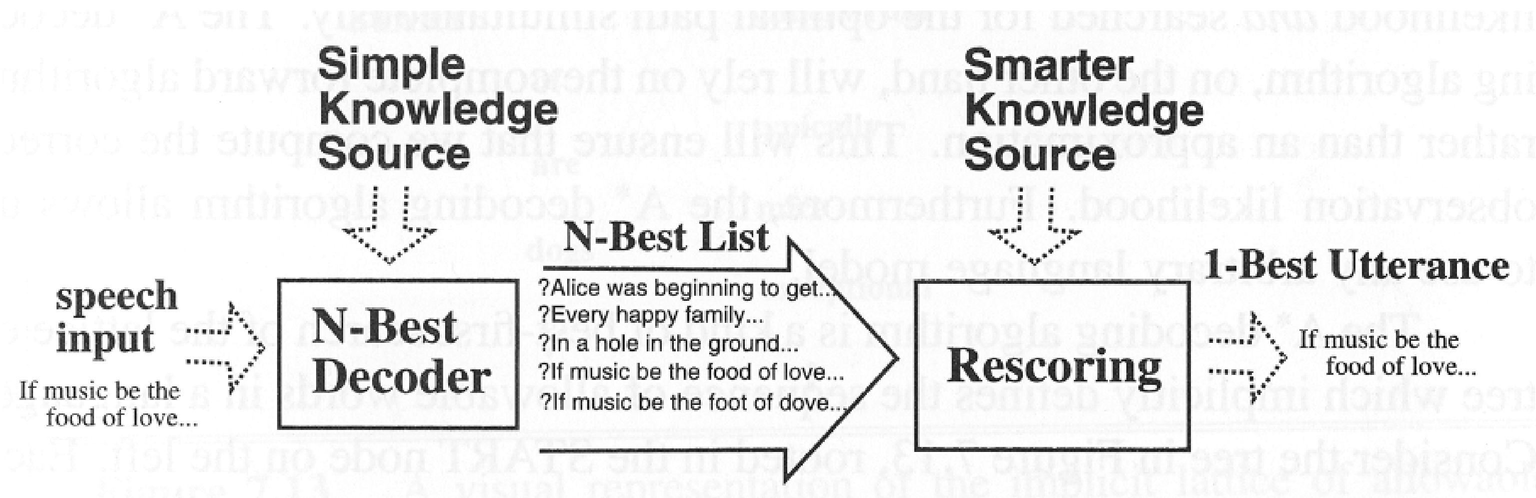


Viterbi decoding of [aa n iy dh ax]:



- Problems
  - Computes most likely *state-sequence* (e.g. phoneme sequence) not *word-sequence*
  - Only for *bigram* grammar models (in our example)
- Solutions
  - Viterbi algorithm as first decoding stage generates
    - *N-best word sequence hypotheses*, or
    - *word lattice* for further decoding stage(s)
  - Tree search: Stack decoder, A* decoder
    - Based on the forward algorithm
    - Best-first search in lattice/tree of sequence of allowable words (in a language)

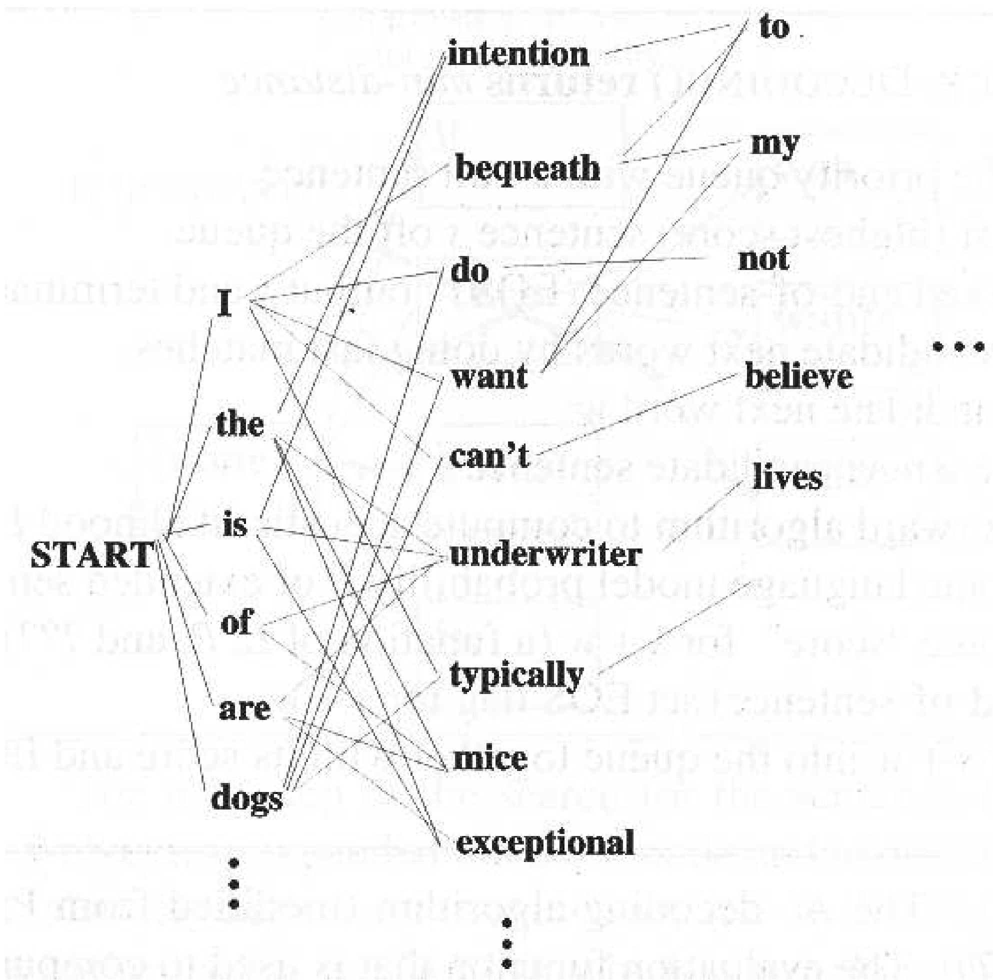
N-best decoding, e.g., as part of a two stage decoder:



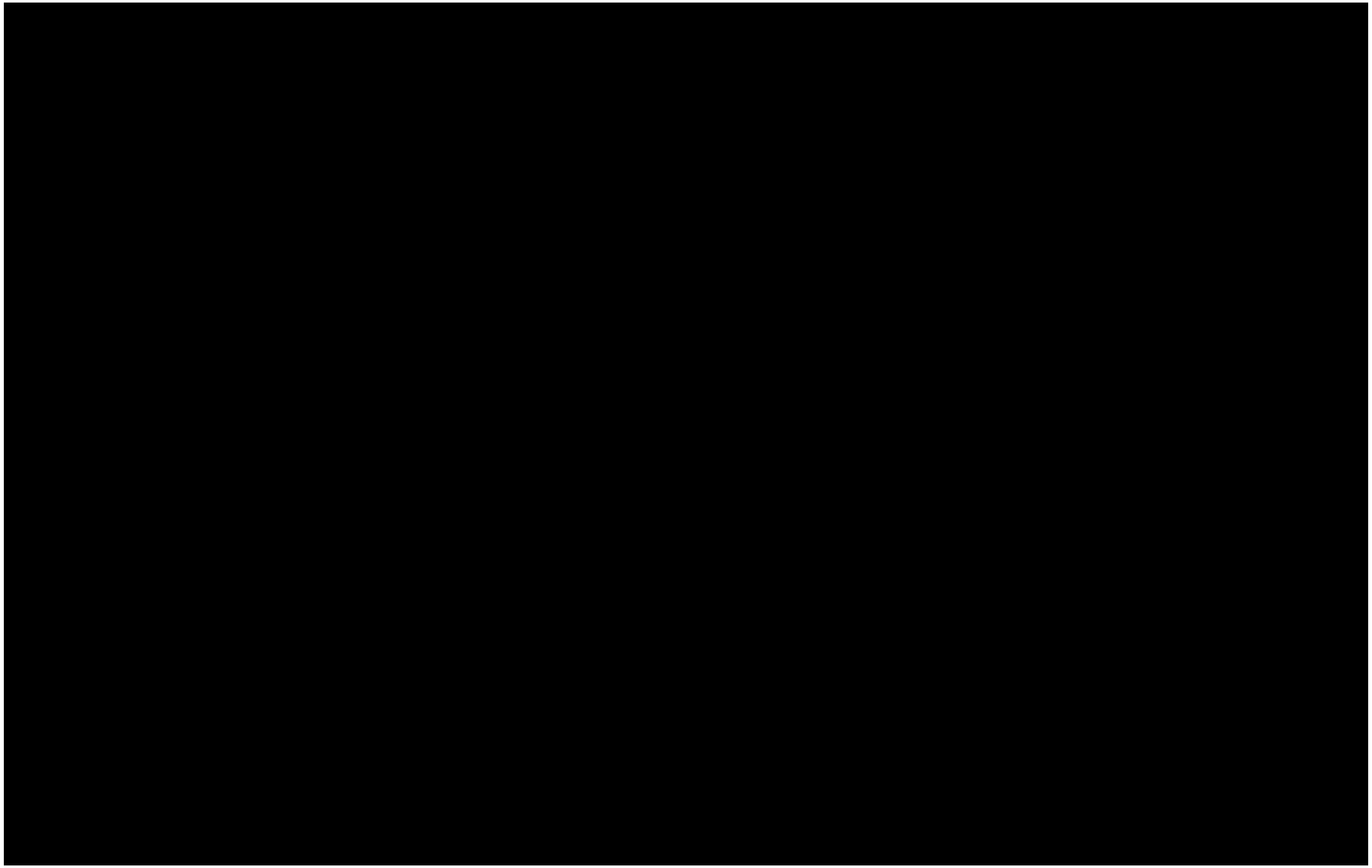
N-Best Decoder:

- Simple, fast
- Reduces search space for second stage

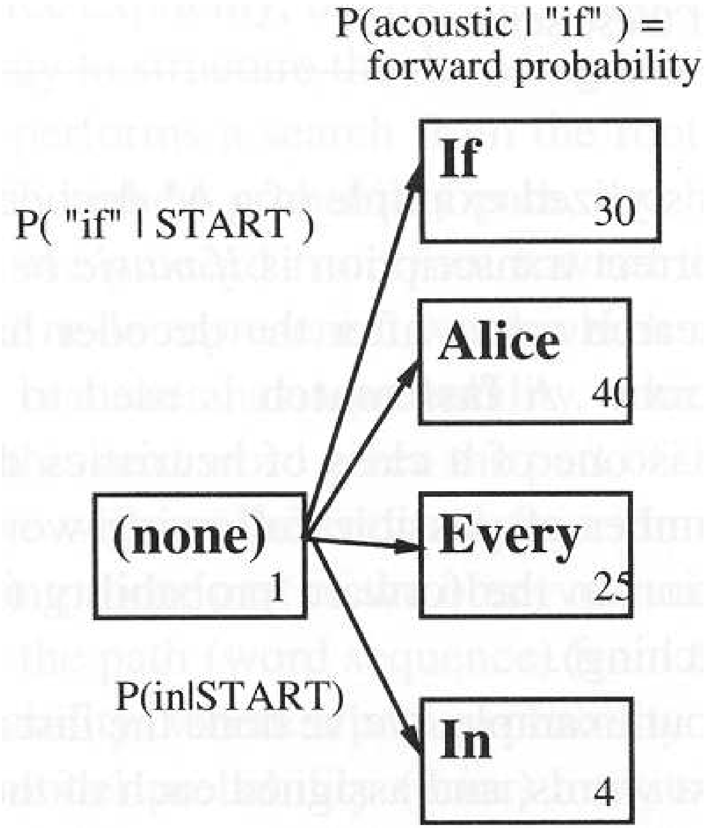
Lattice of allowable word sequences:



Tree Search (Stack Decoding)



Begin of search for the sentence *If music be the food of love.*
At this stage *Alice* is the most likely hypothesis.

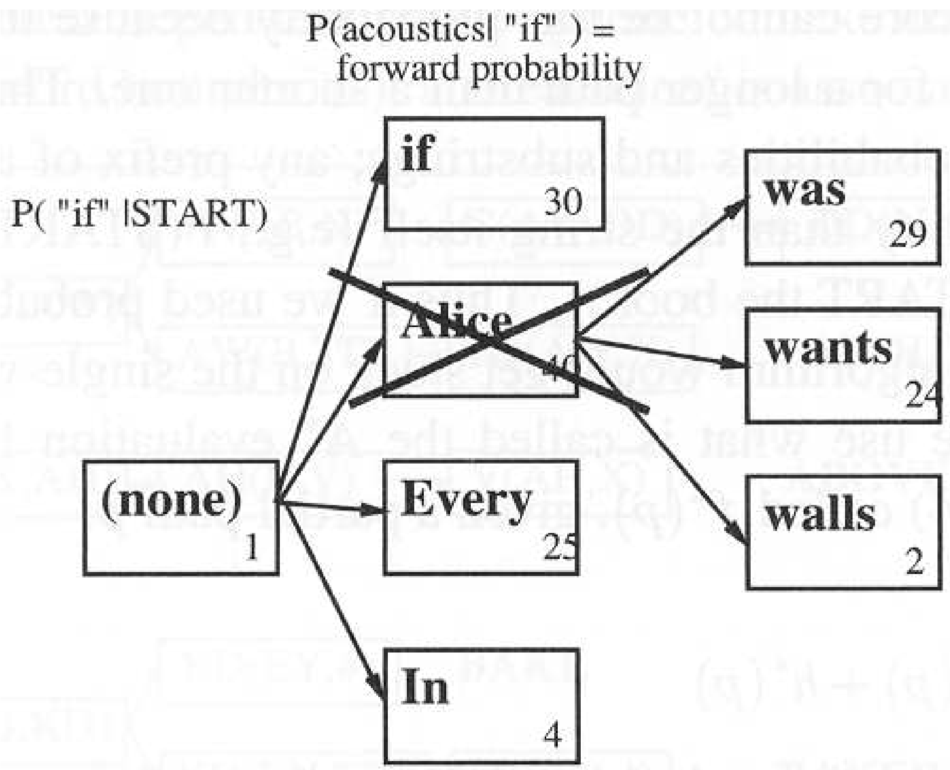


Stack Dekoder (A* Dekoder)

Next step: expand *Alice* node

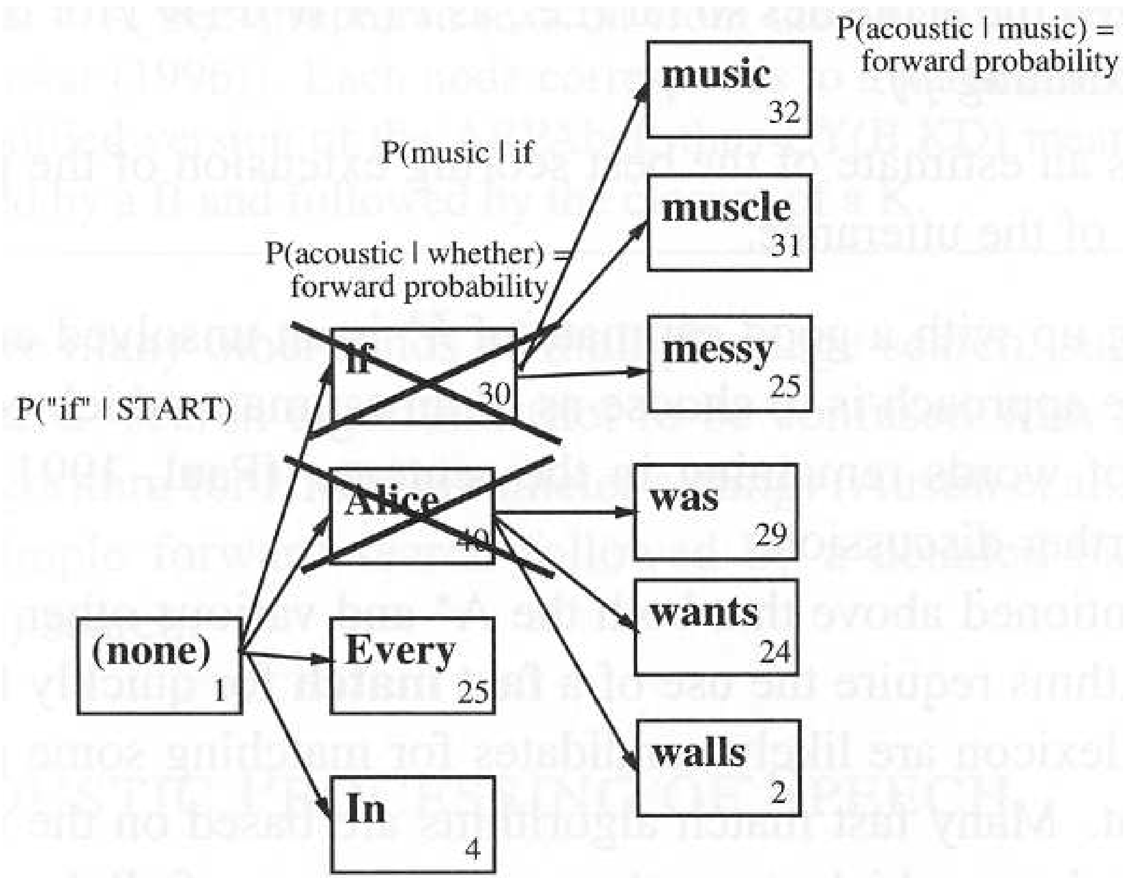
⇒ three extensions with relatively high score

But: highest score for *START if*



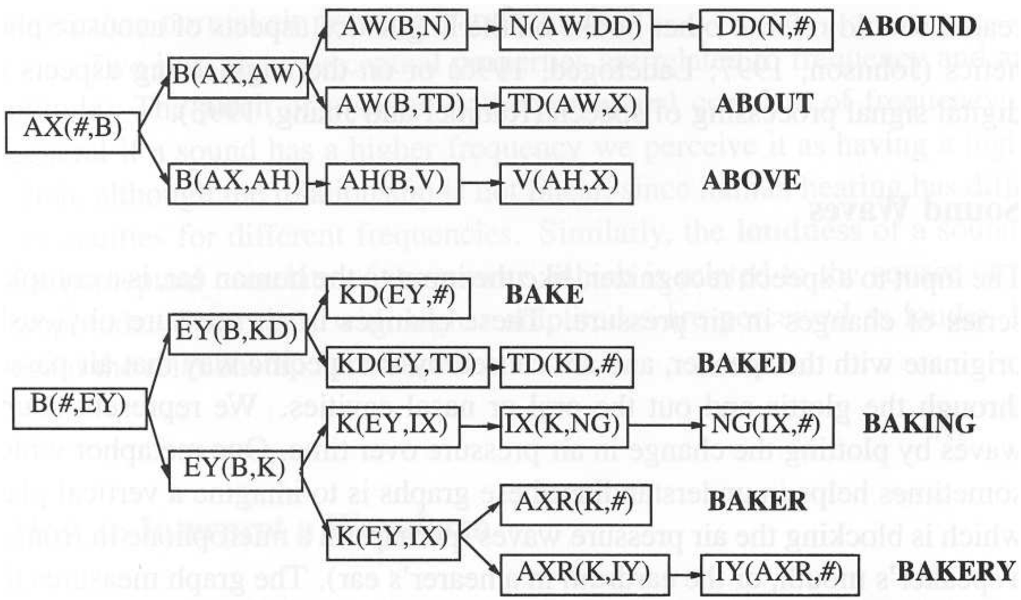
Stack Dekoder (A* Dekoder)

Expand *if* node: highest score for *START if music*



Stack Dekoder (A* Dekoder)

- Score according to probability, $P(X|W)\,P(W)$?
  - Gets smaller for longer word sequences
  - Score has to be extended with estimate of probability for the rest of an utterance
- Use fast match to find extensions for a word, e.g. based on tree structured lexicon:



Zusammenfassung

- Kompiliertes HMM-Netzwerk
  - Viterbi Decoder
  - Vereint akustisches und linguistisches Modell
  - Synchrone Suche
- Tree Search
  - A* Decoder
  - Heuristische Schätzung der Restwahrscheinlichkeiten
  - Asynchrone Suche
- Mehrstufige Dekodierung:
  - N-best Wortliste
  - Wortgitter (lattice)

Referenzen

- E.G. Schukat-Talamazzini, *Automatische Spracherkennung,* Vieweg-Verlag, 1995.
- D. Jurafsky and J.H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice-Hall, 2000.
- F. Jelinek, *Statistical Methods for Speech Recognition*. MIT Press, 1997.