

Vorlesung

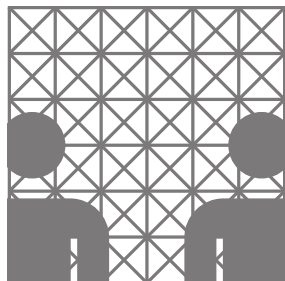
Sprachdialogsysteme

Timo Baumann
baumann@informatik.uni-hamburg.de



<https://nats-www.informatik.uni-hamburg.de/SDS20>

Universität Hamburg, Department of Informatics
Language Technology Group

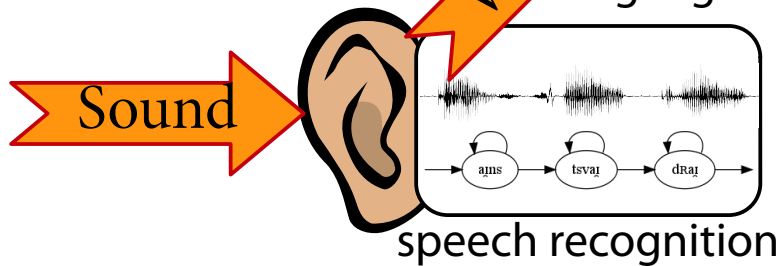
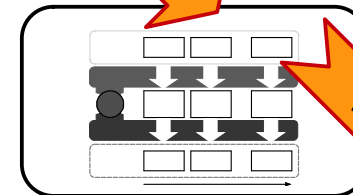
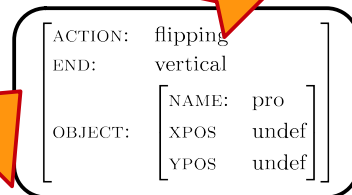


Heute

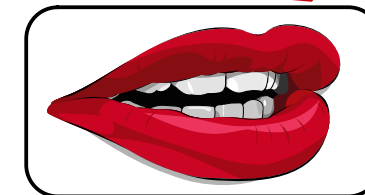
- Reprise zum Dialogmanagement:
Versuch einer Kategorisierung
- kontinuierliche (latente) Zustände
und Ende-zu-Ende-Verarbeitung
= “deep learning”

Ein einfacher Dialogagent

DA = dialog act



language generation

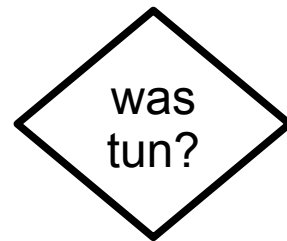


speech synthesis

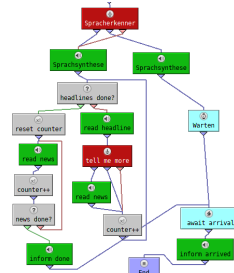


Komponenten im Dialogmanager

Dialogpolicy



Dialogzustand



{
anfang: "A"
geschlecht: f
häufigkeit: normal
anzahl namen: (default)
...
}

Frames

endlicher Automat

Dialog-Manager



Möglichkeiten des Dialogmanagements: Zustände und Aktionsbestimmung

Aktionsbestimmung=Policy

- endlicher Automat
 - ggfs. Erweiterungen durch Skriptsprache
 - Dialogzustand: Zustand im endlichen Automat
 - Aktionsbestimmung: Zustandsübergänge
- Frames
 - Zustand: Inhalt des aktuellen Frames
 - definierte Regeln, wie der Frame weiter zu füllen (oder zu wechseln) ist
 - Regeln sind typischerweise allgemein formuliert (z.B. VoiceXML Form Interpretation Algorithm)
- probabilistisches Dialog-Management
 - Zustände: wie bei Frames
 - Lernen der Policy auf Basis von Interaktionsdaten
 - Policy kann also auf das jeweilige Dialogproblem optimiert werden

probabilistisches Dialog-Management

- probabilistisches Dialog-Management
 - Zustände: wie bei Frames
 - werden also manuell spezifiziert
 - Lernen der Policy auf Basis von Interaktionsdaten
 - welche Aktion ist im jetzigen Zustand wie “gut”?
 - basierend auf Interaktion mit echten oder simulierten Nutzern
 - nur bedingt anhand von (offline gesammelten) Dialogdaten, warum?
 - in welchem Zustand ist das System?
 - MDP: System ist in genau einem Zustand
 - *partially observable* MDP: Wahrscheinlichkeitsverteilung über die möglichen Zustände
 - Aktion auswählen, die gegeben die Wahrscheinlichkeitsverteilung und ihren jeweiligen Nutzen in einem gegebenen Zustand möglichst gut ist

Dialog-Management und (explizite) Nachfragen

- explizite Nachfragen sind immer möglich
 - einerseits: höhere Konfidenz, dass Systemverstehen korrekt ist
 - aber: sehr unnatürlich, verlängern Dialog, können selbst Missverstehen (oder Meinungswechsel) auslösen
- Nachfragen je nach Konfidenz
(der Spracherkennung, der Verstehenskomponente, der Situation, ...)
- Lösung mit POMDPs:
 - unterschiedliche Zustände je nach Konfidenz
 - Wahrscheinlichkeitsverteilung über die Zustände
 - Lernen, wann nachgefragt werden soll, wann nicht
- System fragt dann nach, wenn der erwartete Nutzen höher ist als die erwarteten Kosten (durch längeren Dialog)

Möglichkeiten des Dialogmanagements: was fehlt noch?

- endlicher Automat
 - Dialogzustand: Zustand im endlichen Automat
 - Aktionsbestimmung: Zustandsübergänge
- Frames
 - Zustand: Inhalt des aktuellen Frames
 - definierte Regeln, wie der Frame weiter zu füllen (oder zu wechseln) ist
- probabilistisches Dialog-Management
 - Zustände: wie bei Frames
 - Lernen der Policy auf Basis von Interaktionsdaten

Möglichkeiten des Dialogmanagements: was fehlt noch?

- endlicher Automat
 - Dialogzustand: Zustand im endlichen Automat
 - Aktionsbestimmung: Zustandsübergänge
- Frames
 - Zustand: Inhalt des aktuellen Frames
 - definierte Regeln, wie der Frame weiter zu füllen (oder zu wechseln) ist
- probabilistisches Dialog-Management
 - Zustände: wie bei Frames
 - Lernen der Policy auf Basis von Interaktionsdaten

→ Lernen auch der Zustände

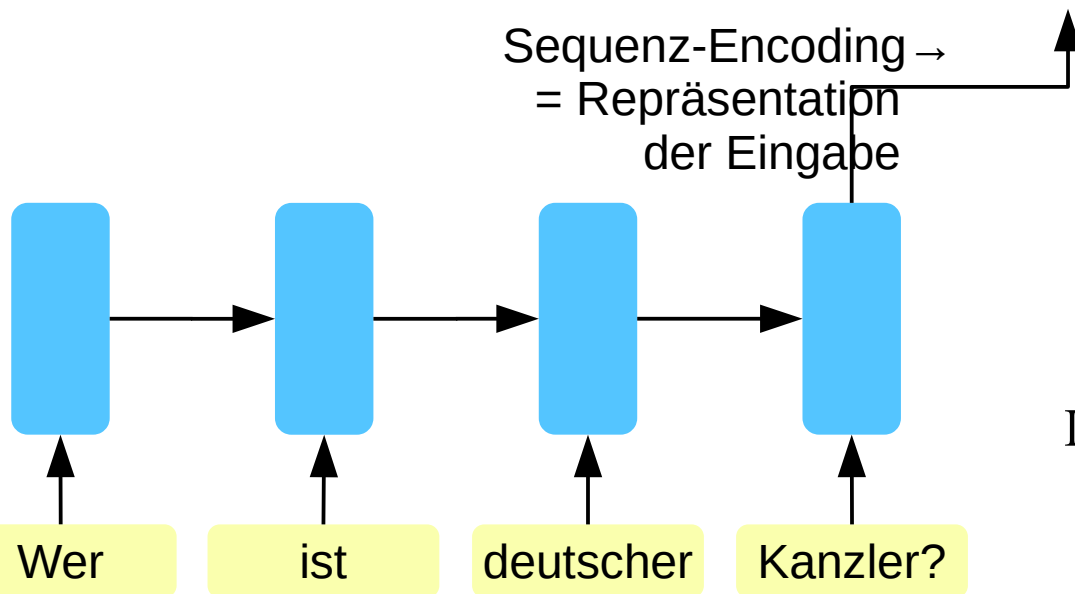
kontinuierliche Repräsentationen in einem Zustandsraum

- bisher:
 - “Zustand” besteht aus diskreten Variablen mit diskreten Werten
 - Variablen und ihren Werten ist eine Bedeutung zugeordnet
z.B.: { Abfahrtsort: Hamburg, Konfidenz: hoch }
 - oder (POMDP):
 - [0.8: { Abfahrtsort: Hamburg, Konfidenz: hoch },
 - 0.1: { Abfahrtsort: Hamburg, Konfidenz: niedrig },
 - 0.1: { Abfahrtsort: Bad Homburg, Konfidenz: hoch }, ...]
- Idee:
 - Zustand besteht aus kontinuierlichen Variablen
 - n Variable \rightarrow n -dimensionaler *Zustandsraum*
 - Dialogzustand \rightarrow Punkt im Zustandsraum
 - ähnliche Zustände \rightarrow räumliche Nähe

Frage-Antwort-Systeme und Sequence-to-Sequence Learning

- vergleichbar zu Übersetzung:
 - “Quellsprache”: Fragen
 - “Zielsprache”: Antworten
 - erfolgreiche Übersetzung: passende Antwort zur Frage
- große Menge an Frage-Antwort-Paaren verfügbar für viele Domänen (frag-mutti.de, ...)
- Vorgehen:
 - “Encodierung” der Frage in den Zustandsraum
 - Suche der passenden Antwort zu diesem Punkt im Zustandsraum
 - häufig “Dekodierung”, stückweises

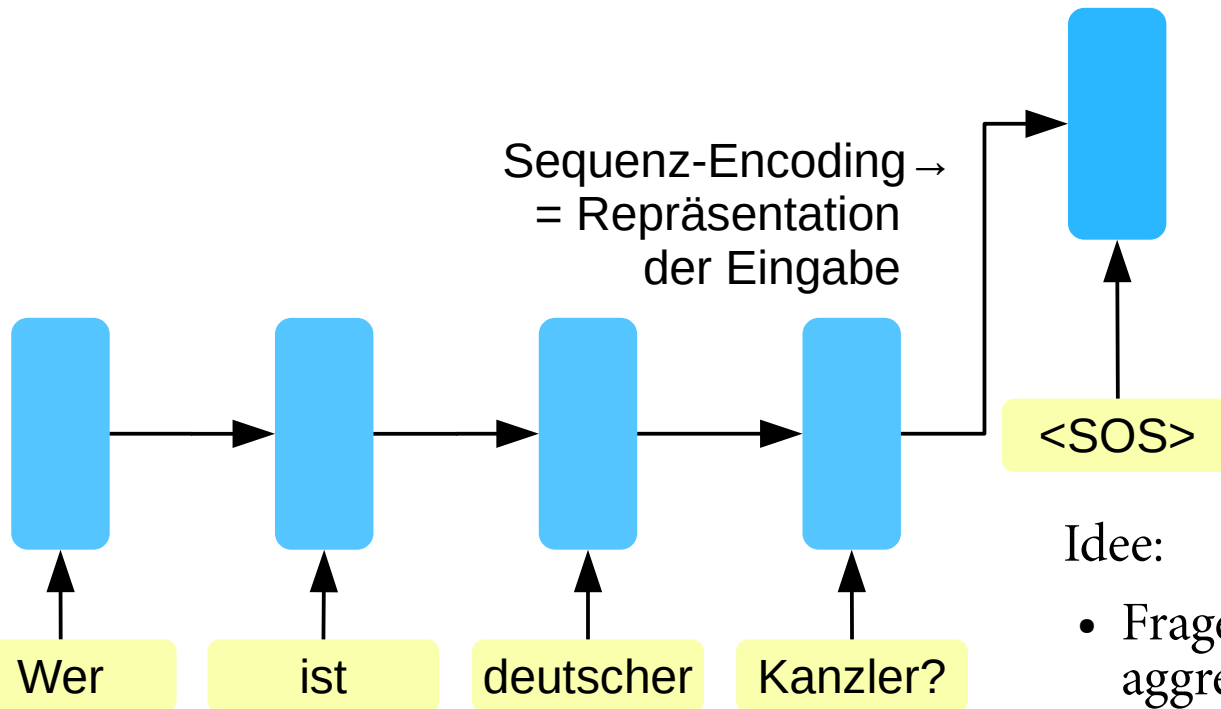
Encoder-Decoder-Architektur



Idee:

- Frage-Information wird im Zustand aggregiert
- Zustand wird stück-für-stück zur Antwort dekodiert
- Lernen aus Frage-Antwort-Paaren ...
- ... mittels neuronaler Netze

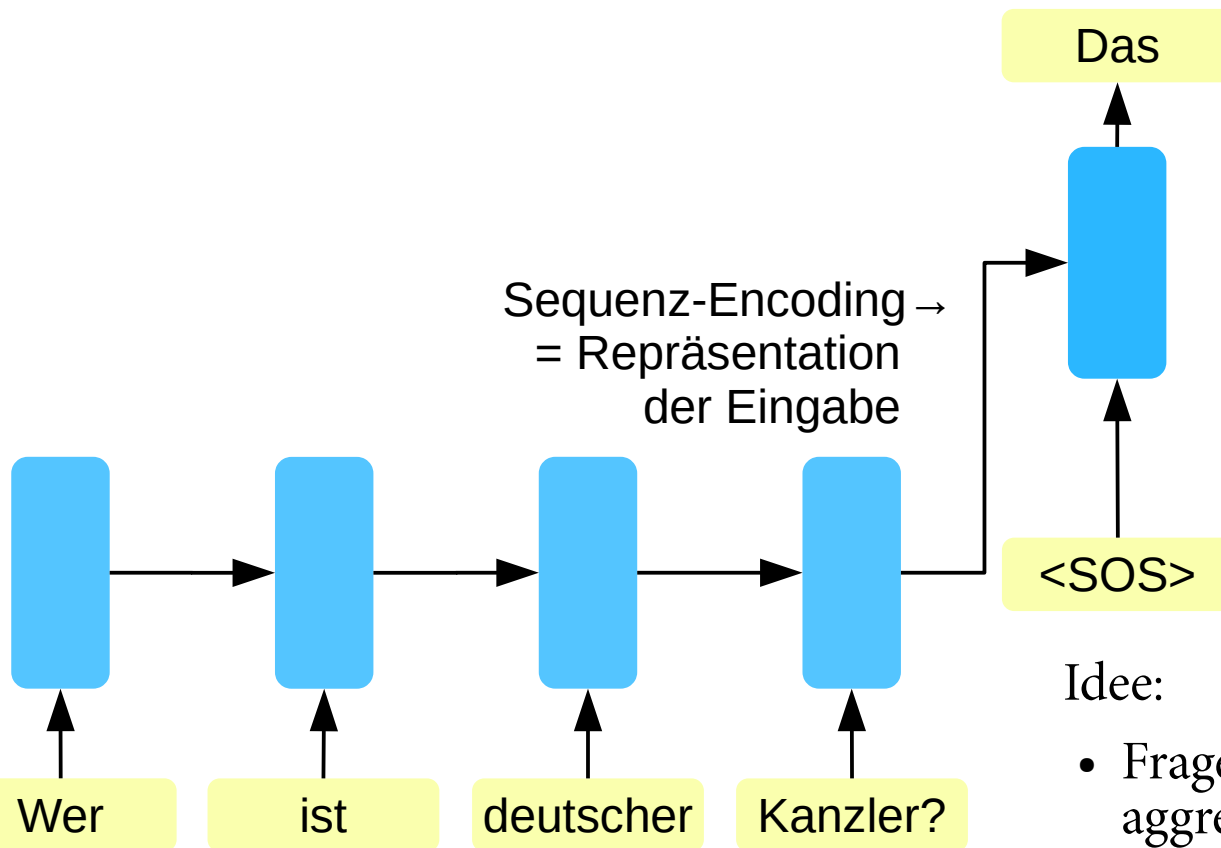
Encoder-Decoder-Architektur



Idee:

- Frage-Information wird im Zustand aggregiert
- Zustand wird stück-für-stück zur Antwort dekodiert
- Lernen aus Frage-Antwort-Paaren ...
- ... mittels neuronaler Netze

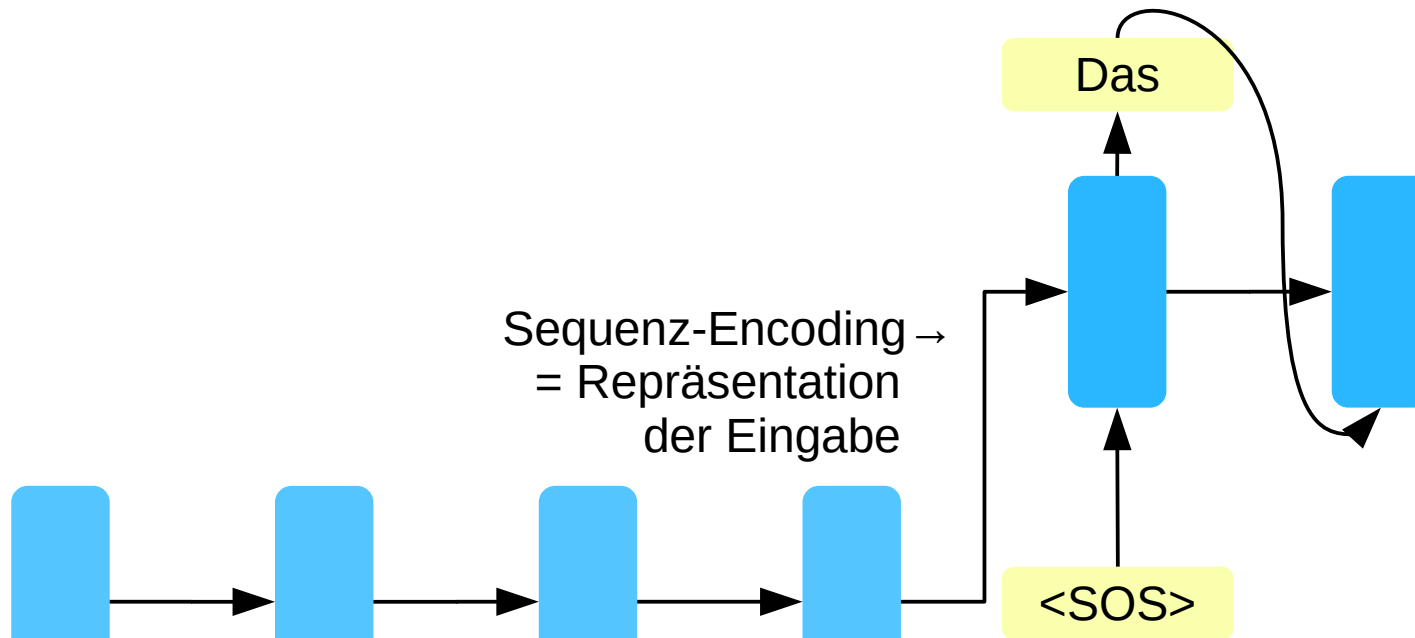
Encoder-Decoder-Architektur



Idee:

- Frage-Information wird im Zustand aggregiert
- Zustand wird stück-für-stück zur Antwort dekodiert
- Lernen aus Frage-Antwort-Paaren ...
- ... mittels neuronaler Netze

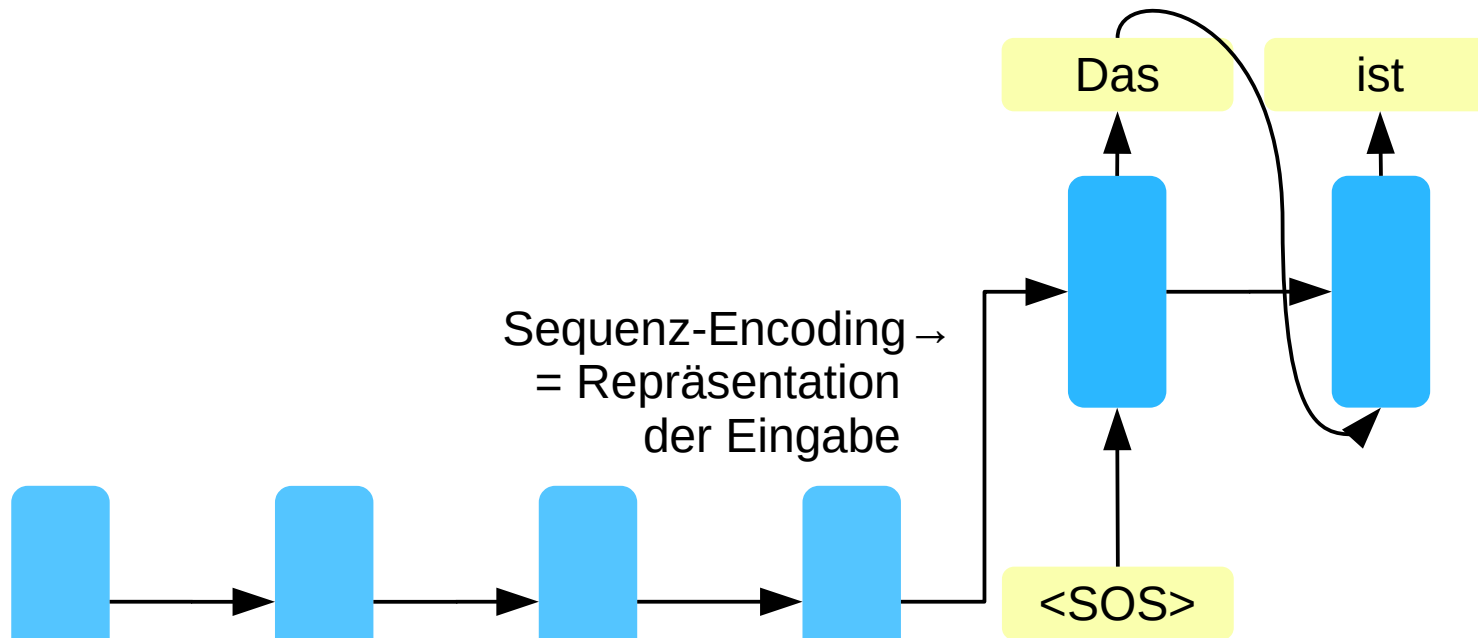
Encoder-Decoder-Architektur



Idee:

- Frage-Information wird im Zustand aggregiert
- Zustand wird stück-für-stück zur Antwort dekodiert
- Lernen aus Frage-Antwort-Paaren ...
- ... mittels neuronaler Netze

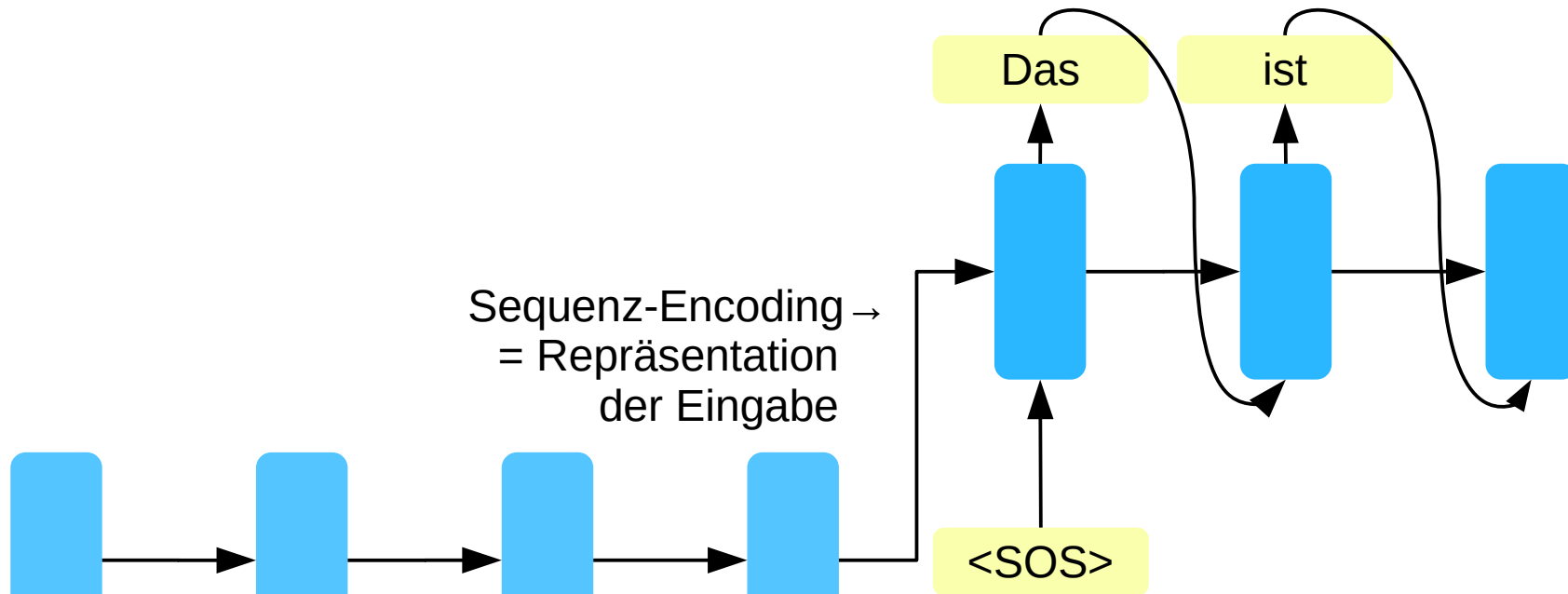
Encoder-Decoder-Architektur



Idee:

- Frage-Information wird im Zustand aggregiert
- Zustand wird stück-für-stück zur Antwort dekodiert
- Lernen aus Frage-Antwort-Paaren ...
- ... mittels neuronaler Netze

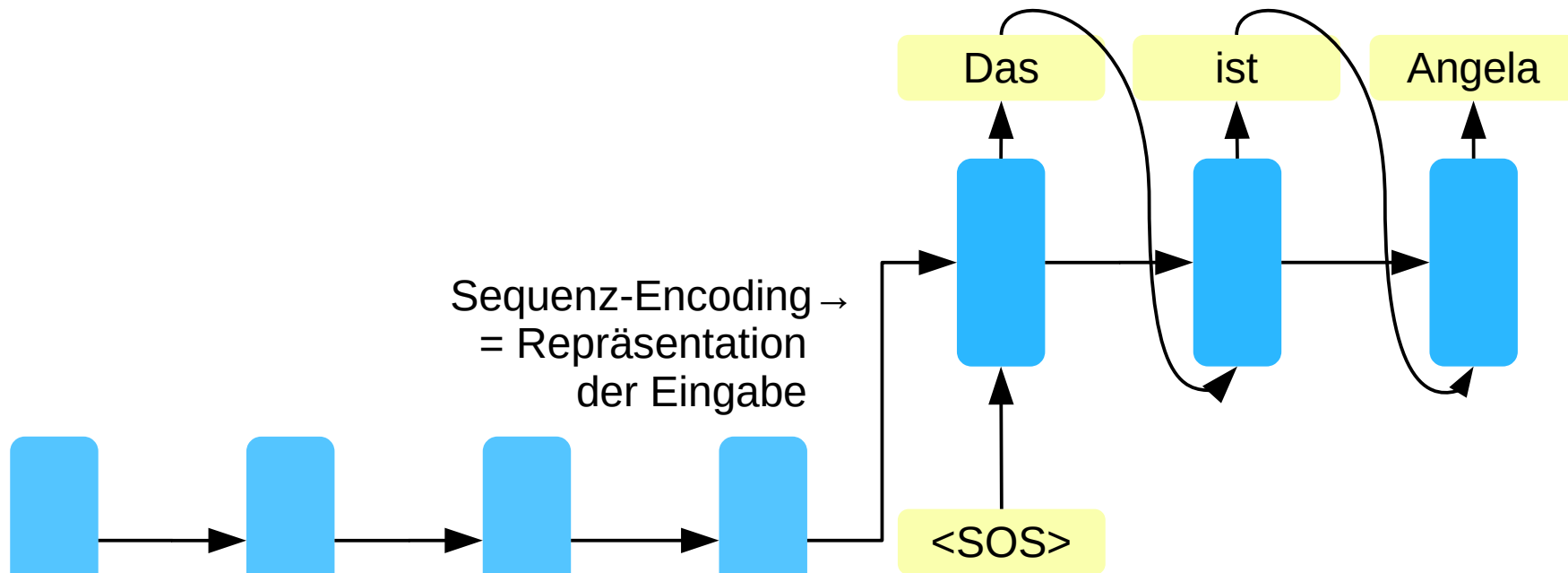
Encoder-Decoder-Architektur



Idee:

- Frage-Information wird im Zustand aggregiert
- Zustand wird stück-für-stück zur Antwort dekodiert
- Lernen aus Frage-Antwort-Paaren ...
- ... mittels neuronaler Netze

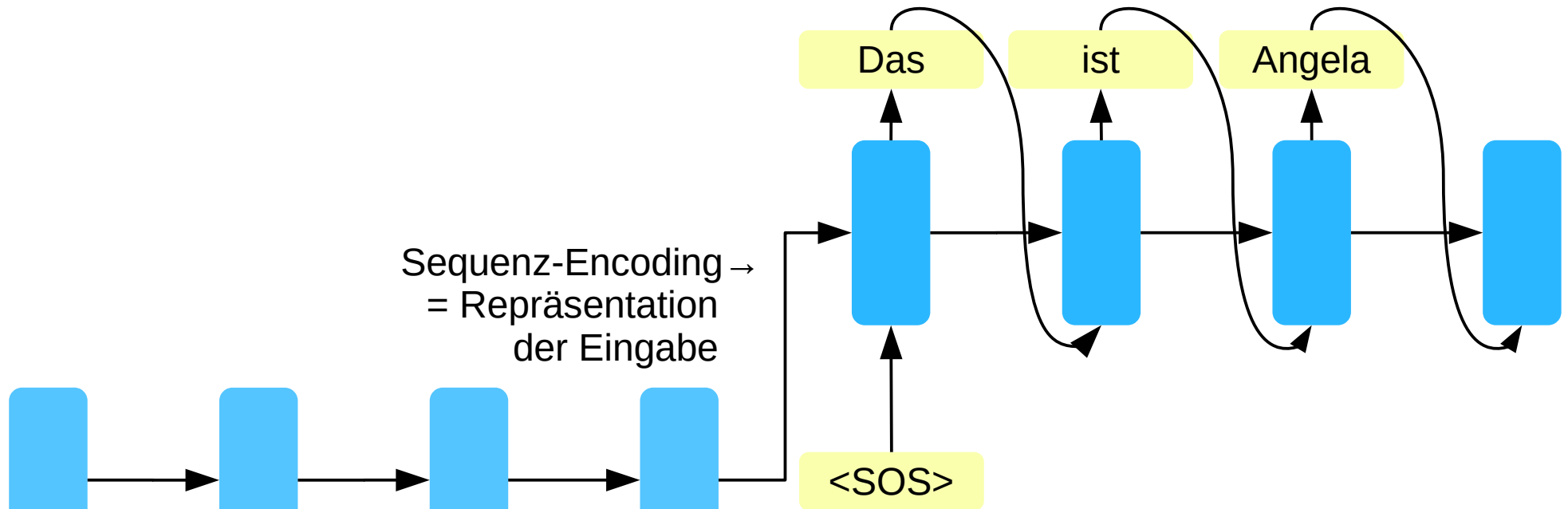
Encoder-Decoder-Architektur



Idee:

- Frage-Information wird im Zustand aggregiert
- Zustand wird stück-für-stück zur Antwort dekodiert
- Lernen aus Frage-Antwort-Paaren ...
- ... mittels neuronaler Netze

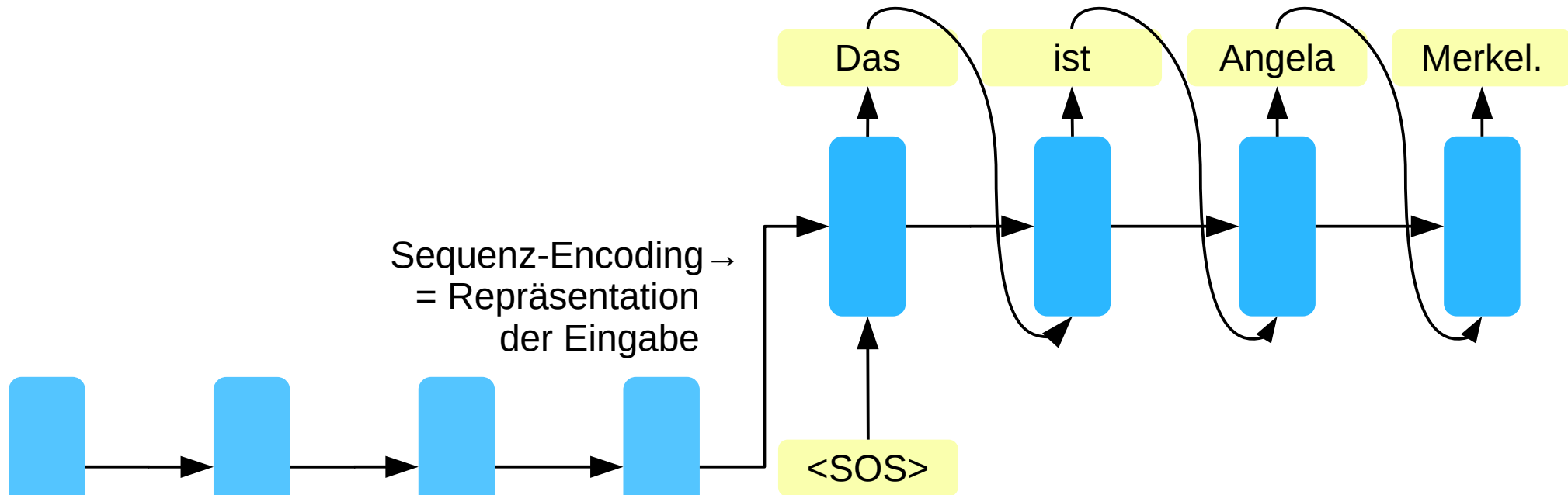
Encoder-Decoder-Architektur



Idee:

- Frage-Information wird im Zustand aggregiert
- Zustand wird stück-für-stück zur Antwort dekodiert
- Lernen aus Frage-Antwort-Paaren ...
- ... mittels neuronaler Netze

Encoder-Decoder-Architektur



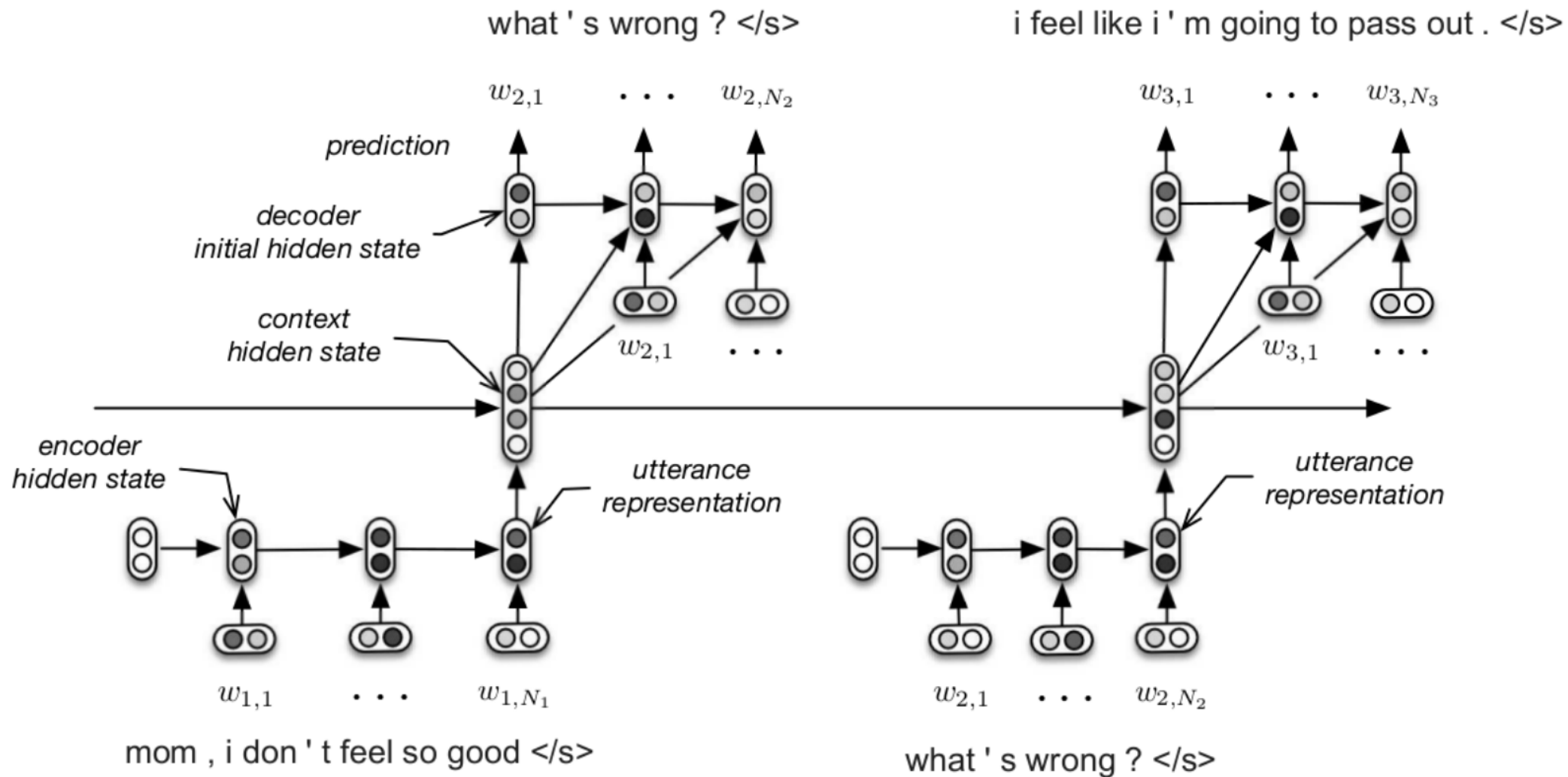
Idee:

- Frage-Information wird im Zustand aggregiert
- Zustand wird stück-für-stück zur Antwort dekodiert
- Lernen aus Frage-Antwort-Paaren ...
- ... mittels neuronaler Netze

neuronales Dialogmanagement

- Lernen sowohl des Zustandsraums, als auch der Aktionen
- meistens: “end-to-end”, also von Eingabe(text) zu Ausgabe(text), ohne explizite semantische Ebene

Verstehen, Repräsentieren und Generieren in einem



neuronales Dialogmanagement

- Lernen sowohl des Zustandsraums, als auch der Aktionen
- meistens: “end-to-end”, also von Eingabe(text) zu Ausgabe(text), ohne explizite semantische Ebene
- sehr aktives Forschungsfeld
- derzeit noch eher bescheidene Ergebnisse

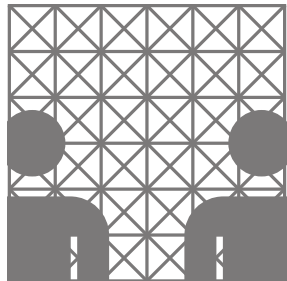
Vielen Dank.

baumann@informatik.uni-hamburg.de



<https://nats-www.informatik.uni-hamburg.de/SDS20>

Universität Hamburg, Department of Informatics
Language Technology Group



Notizen