CS447: Natural Language Processing

http://courses.engr.illinois.edu/cs447

Lecture 27
Seq2Seq, Attention;
Generation and Dialog

Julia Hockenmaier

juliahmr@illinois.edu

3324 Siebel Center

Final exam

Wednesday, Dec 12 in class
Only materials after midterm
Same format as midterm

Review session this Friday!

Where we're at

Lecture 25: Word Embeddings and neural LMs

Lecture 26: Recurrent networks and Sequence Labeling

Lecture 27: Seq2Seq, Attention, Generation and Dialog

Lecture 28: Review for the final exam

Lecture 29: In-class final exam

Today's lecture

Traditional NLG and traditional dialogue systems
very quick overview

The workhorse behind current neural approaches:
seq2seq models with attention

Traditional NLG…

What is Generation?

Automatic production of natural language text, usually from underlying semantic representation

- As “natural-language front ends” used to present information in databases etc.:
  weather forecasts, train systems,
  (personalized) museum/restaurant/shopping guides,…
- In dialog systems
- In summarization systems
- In authoring aids to help people create routine documents: customer support, job ads, etc…

Example: Rail travel information system

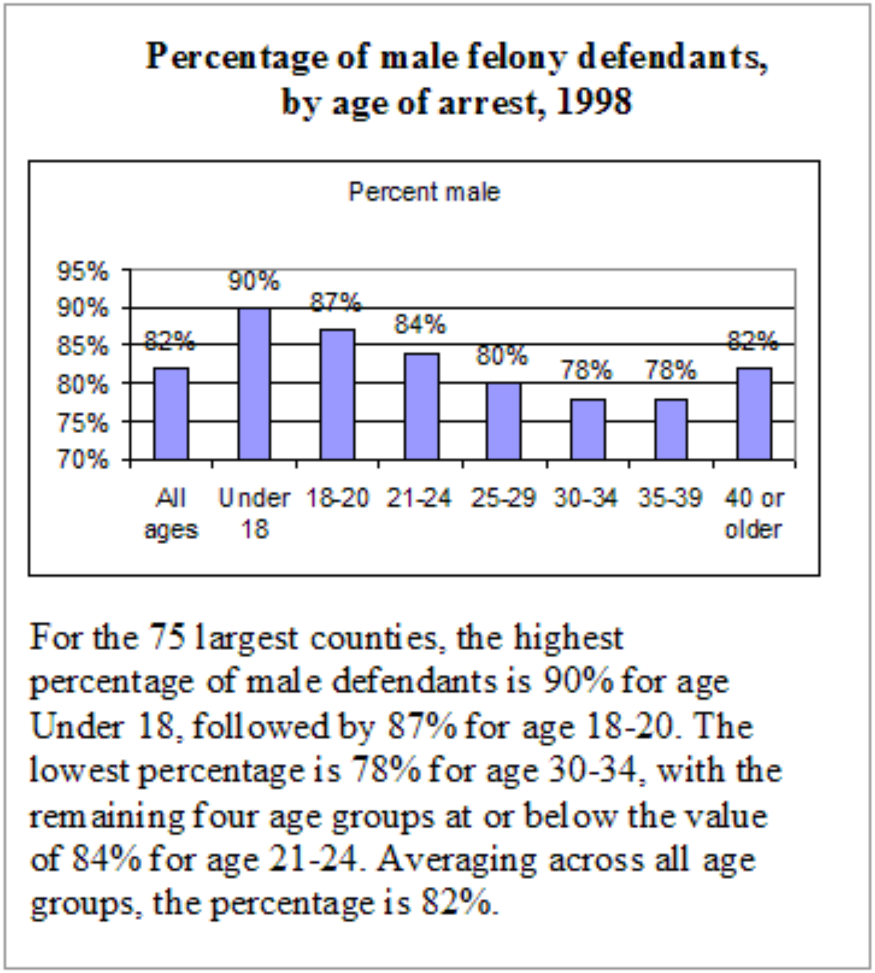
- **Domain knowledge:** Train schedules
- **User Input:** from a graphical user interface, or in natural language: *"How can I get from Aberdeen to Glasgow?"*

- **Desired output:**
There are 20 trains each day from Aberdeen to Glasgow. The next train is the Caledonian Express; it leaves Aberdeen at 10am. It is due to arrive in Glasgow at 1pm, but arrival may be slightly delayed because of snow on the track near Stirling.

Some NLG systems

Cogentex's chart explainer

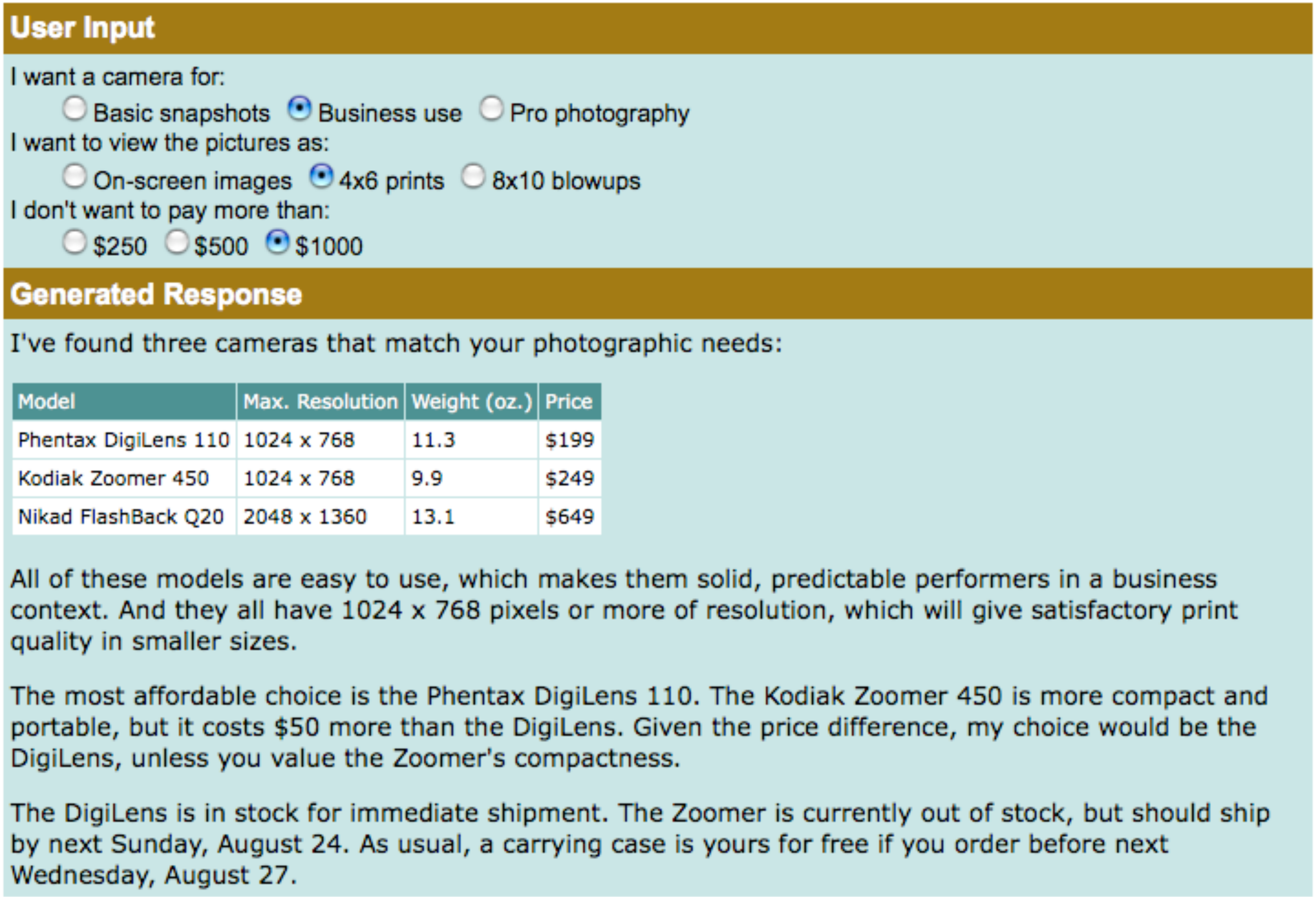
http://www.cogentex.com/products/chartex/faq/bjs-sample.png



Percentage of male felony defendants, by age of arrest, 1998

For the 75 largest counties, the highest percentage of male defendants is 90% for age Under 18, followed by 87% for age 18-20. The lowest percentage is 78% for age 30-34, with the remaining four age groups at or below the value of 84% for age 21-24. Averaging across all age groups, the percentage is 82%.

Cogentex's Camera system



User Input

I want a camera for:

○ Basic snapshots ● Business use ○ Pro photography

I want to view the pictures as:

○ On-screen images ● 4x6 prints ○ 8x10 blowups

I don't want to pay more than:

○ $250 ○ $500 ● $1000

Generated Response

I've found three cameras that match your photographic needs:

Model	Max. Resolution	Weight (oz.)	Price
Phentax DigiLens 110	1024 x 768	11.3	$199
Kodiak Zoomer 450	1024 x 768	9.9	$249
Nikad FlashBack Q20	2048 x 1360	13.1	$649

All of these models are easy to use, which makes them solid, predictable performers in a business context. And they all have 1024 x 768 pixels or more of resolution, which will give satisfactory print quality in smaller sizes.

The most affordable choice is the Phentax DigiLens 110. The Kodiak Zoomer 450 is more compact and portable, but it costs $50 more than the DigiLens. Given the price difference, my choice would be the DigiLens, unless you value the Zoomer's compactness.

The DigiLens is in stock for immediate shipment. The Zoomer is currently out of stock, but should ship by next Sunday, August 24. As usual, a carrying case is yours for free if you order before next Wednesday, August 27.

Edinburgh’s ILEX and M-PIRO

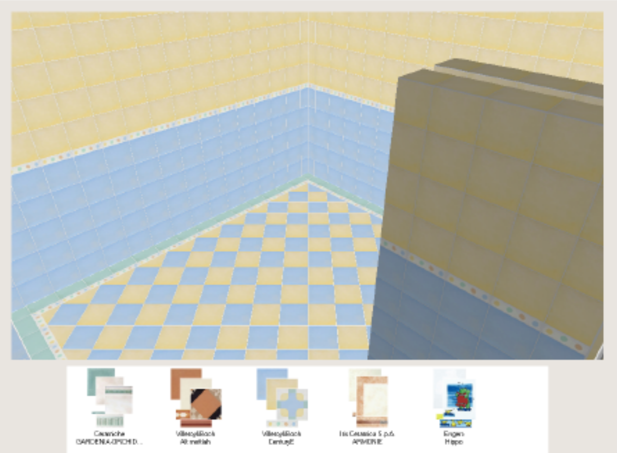
ILEX: a web-based virtual museum gallery and a phone-based system for an actual gallery
M-PIRO: adds an authoring tool for curators

What is that?

This exhibit is a lekythos, created during the archaic period. It dates from circa 500 BC. It was painted by Amasis with the red figure technique and it originates from Attica.

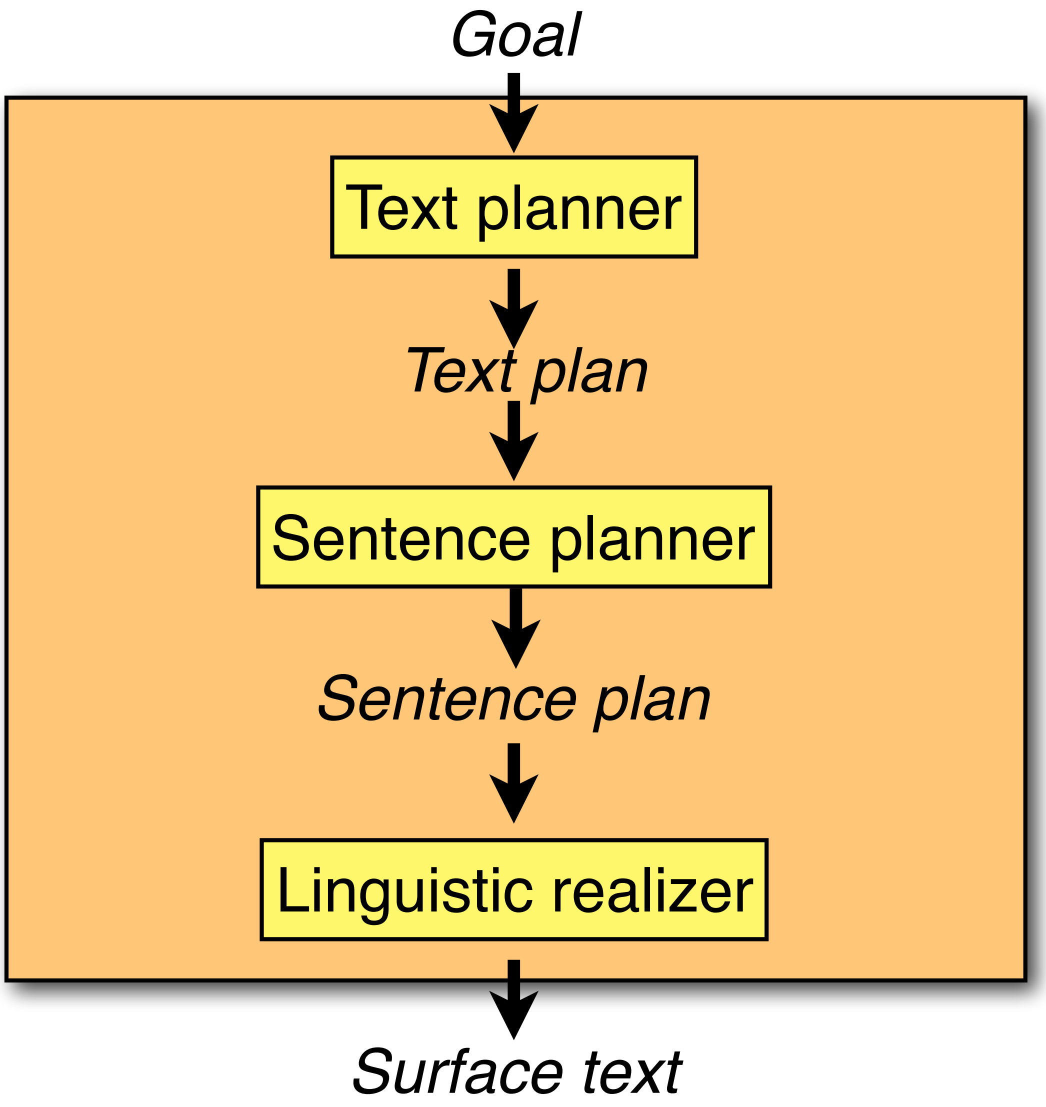
The COMIC system

Conversational **M**ultimodal **I**nteraction with **C**omputers
Dialog system for bathroom design applications


Ceramiche GARDENIA-ORCHID...
Villeroy&Boch Alt mettlach
Villeroy&Boch CenturyE
Iris Ceramica S.p.A. ARMONIE
Engers Hippo

User	Tell me about this design *[click on Alt Mettlach]*
COMIC	*[Look at screen]* THIS DESIGN is in the CLASSIC style. *[circle tiles]* As you can see, the colours are DARK RED and OFF WHITE. *[point at tiles]* The tiles are from the ALT METTLACH collection by VILLEROY AND BOCH. *[point at design name]*

NLG architecture



Goal

Text planner

Text plan

Sentence planner

Sentence plan

Linguistic realizer

Surface text

NLG architectures

There are many dependencies between these tasks. The standard NLG system architecture consists of:

Text planning:

Content determination and discourse planning

Sentence planning:

Sentence aggregation, lexicalization and referring expression generation

Linguistic realization:

Syntactic, morphological and orthographic processing.

NLG tasks

Text planning

1. Content determination

What information (what ‘messages’) should be communicated?

2. Discourse planning

How should the messages be structured/ordered?

Sentence planning

3. Sentence aggregation

Which messages should be combined into individual sentences?

4. Lexicalization

In which words/phrases should domain concepts/relations be expressed?

5. Referring expression generation

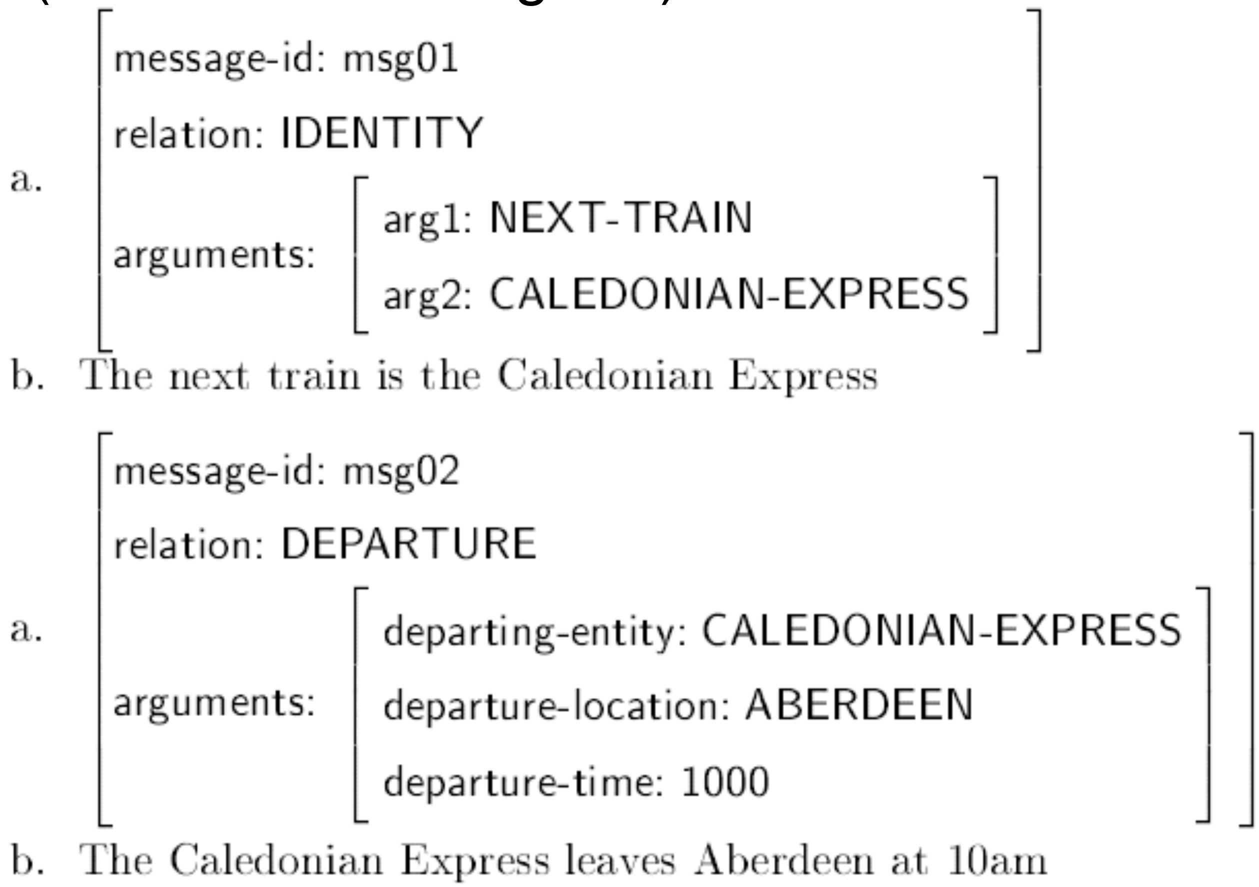
How should entities be referred to?

6. Linguistic realization

Generate a grammatical and orthographically well-formed text

Content determination

Input: user input and background knowledge (database)
Output: a set of ‘messages’ to be communicated
(here shown with gloss)



a.
$$\begin{bmatrix} \text{message-id: msg01} & \\ \text{relation: IDENTITY} & \\ \text{arguments:} & \begin{bmatrix} \text{arg1: NEXT-TRAIN} \\ \text{arg2: CALEDONIAN-EXPRESS} \end{bmatrix} \end{bmatrix}$$

b. The next train is the Caledonian Express

a.
$$\begin{bmatrix} \text{message-id: msg02} & \\ \text{relation: DEPARTURE} & \\ \text{arguments:} & \begin{bmatrix} \text{departing-entity: CALEDONIAN-EXPRESS} \\ \text{departure-location: ABERDEEN} \\ \text{departure-time: 1000} \end{bmatrix} \end{bmatrix}$$

b. The Caledonian Express leaves Aberdeen at 10am

Content determination

Input: user input and background knowledge (database)
Output: a set of 'messages' to be communicated

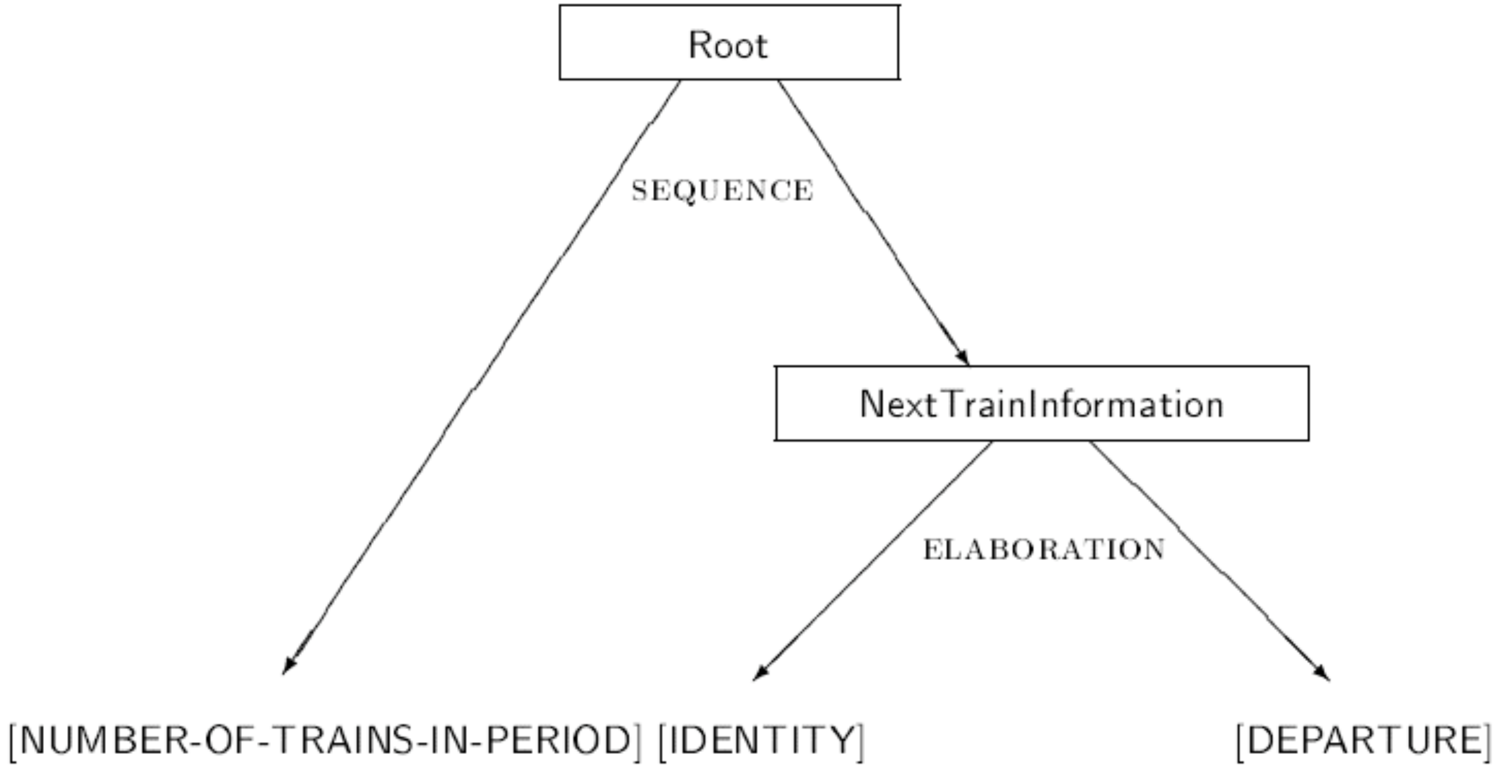
User model: User's task, user's level of expertise, previous interactions with system (esp. in dialog)

Need to filter, summarize and process input data

Relies often on (system-specific) heuristics
(looking at corpus helps!)

Discourse planning

How should the messages be ordered?
What are the discourse relations that hold between them?
Often represented as a tree:



Sentence aggregation

Which messages should be conveyed in a single sentence?

The next train leaves at 10am. It is the Caledonian Express.

The next train, which leaves at 10am, is the Caledonian Express.

Linguistic means to combine messages (=clauses):

- **Relative clauses:** *The next train, which leaves at 10 am, is the Caledonian Express*
- **Coordination:** *The Caledonian Express leaves at 10am, and is the next train*
- **Subordination:** *The Caledonian Express is the next train, although it leaves only at 10am.*
- **Lists:** *There are trains at 10am, at 11:30am and at 1:00pm.*

Lexicalization and referring expressions

Lexicalization: Which words and phrases should be used to express domain concepts:

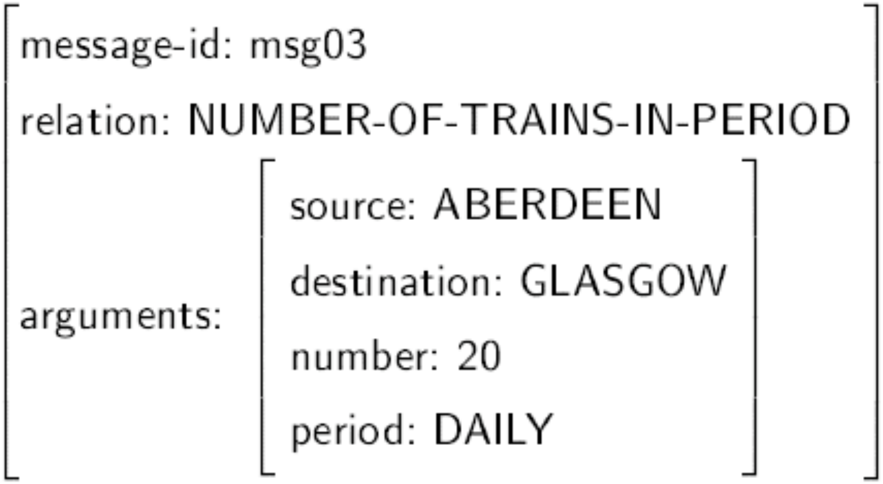
- does the train ‘leave’ or ‘depart’?
- a ‘statistical error’ is not the same as a ‘statistical mistake’

NLG systems need a domain lexicon

Referring expression generation:
When do you use a pronoun/a definite NP/an indefinite NP to refer to an entity?
Needs a discourse model

Linguistic realization

Generate a grammatically and orthographically correct English utterance:



message-id: msg03

relation: NUMBER-OF-TRAINS-IN-PERIOD

arguments:

source: ABERDEEN

destination: GLASGOW

number: 20

period: DAILY

There are *20 train**s each*** *day* ***from*** *Glasgow* ***to*** *Edinburgh.*

NLG evaluation

Many areas of NLP have shared task evaluations that allow comparisons of different algorithms/systems on the same data.

But most NLG systems are very domain/application specific.

- Every system starts from its own input representation
- Not a single gold standard data set
- Can we evaluate subtasks (e.g. referring expression generation)?
- How can we compare system outputs against each other/ against human produced text?
  (metrics such as BLEU/ROUGE may not correlate highly enough with human judgments)

Conversational Agents (Chapter 24)

Conversational Agents

Systems that are capable of performing a task-driven dialog with a human user.

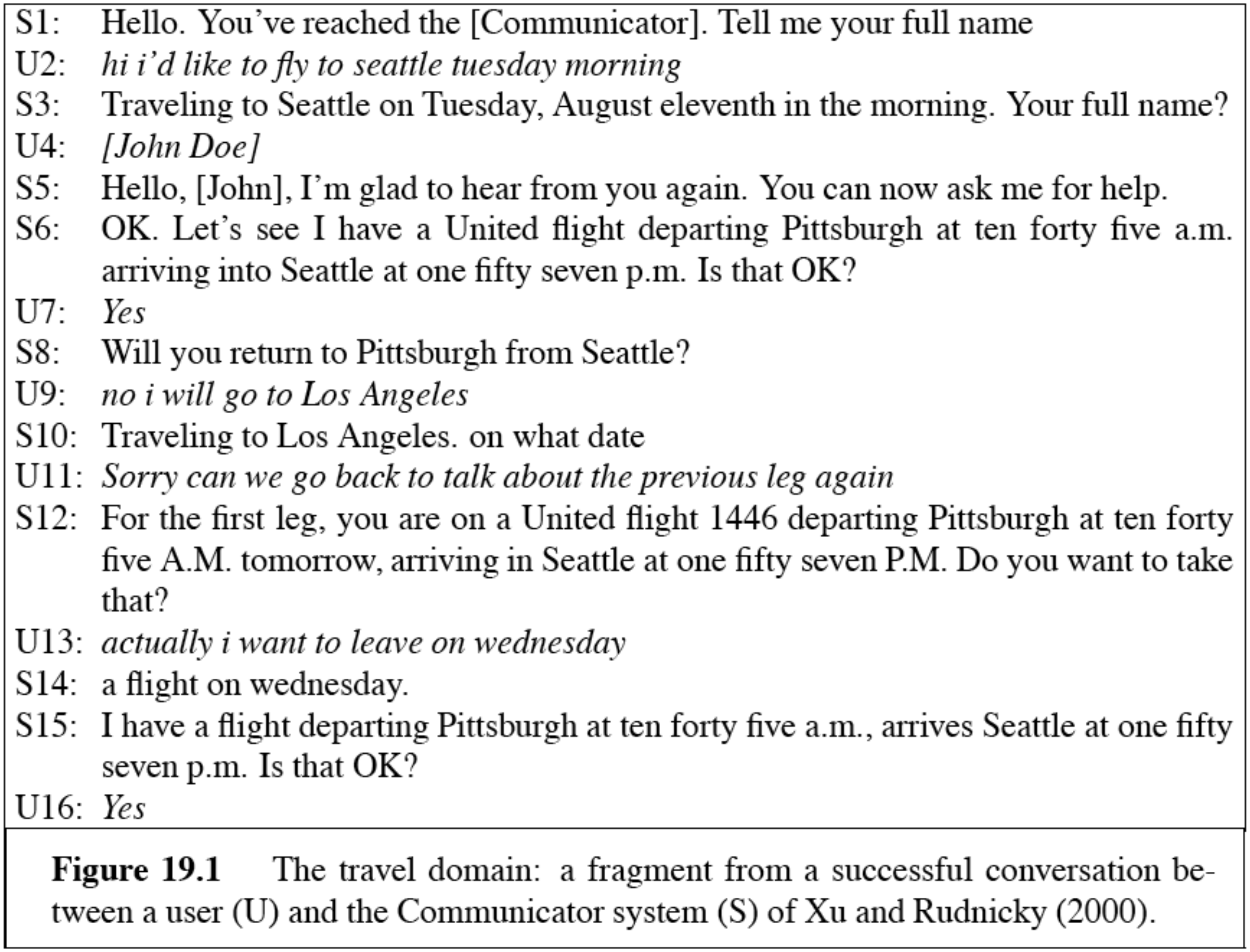
AKA:

- Spoken Language Systems
- Dialogue Systems
- Speech Dialogue Systems

Applications:

- Travel arrangements (Amtrak, United airlines)
- Telephone call routing
- Tutoring
- Communicating with robots
- Anything with limited screen/keyboard

A travel dialog: Communicator



S1: Hello. You've reached the [Communicator]. Tell me your full name

U2: *hi i'd like to fly to seattle tuesday morning*

S3: Traveling to Seattle on Tuesday, August eleventh in the morning. Your full name?

U4: *[John Doe]*

S5: Hello, [John], I'm glad to hear from you again. You can now ask me for help.

S6: OK. Let's see I have a United flight departing Pittsburgh at ten forty five a.m. arriving into Seattle at one fifty seven p.m. Is that OK?

U7: *Yes*

S8: Will you return to Pittsburgh from Seattle?

U9: *no i will go to Los Angeles*

S10: Traveling to Los Angeles. on what date

U11: *Sorry can we go back to talk about the previous leg again*

S12: For the first leg, you are on a United flight 1446 departing Pittsburgh at ten forty five A.M. tomorrow, arriving in Seattle at one fifty seven P.M. Do you want to take that?

U13: *actually i want to leave on wednesday*

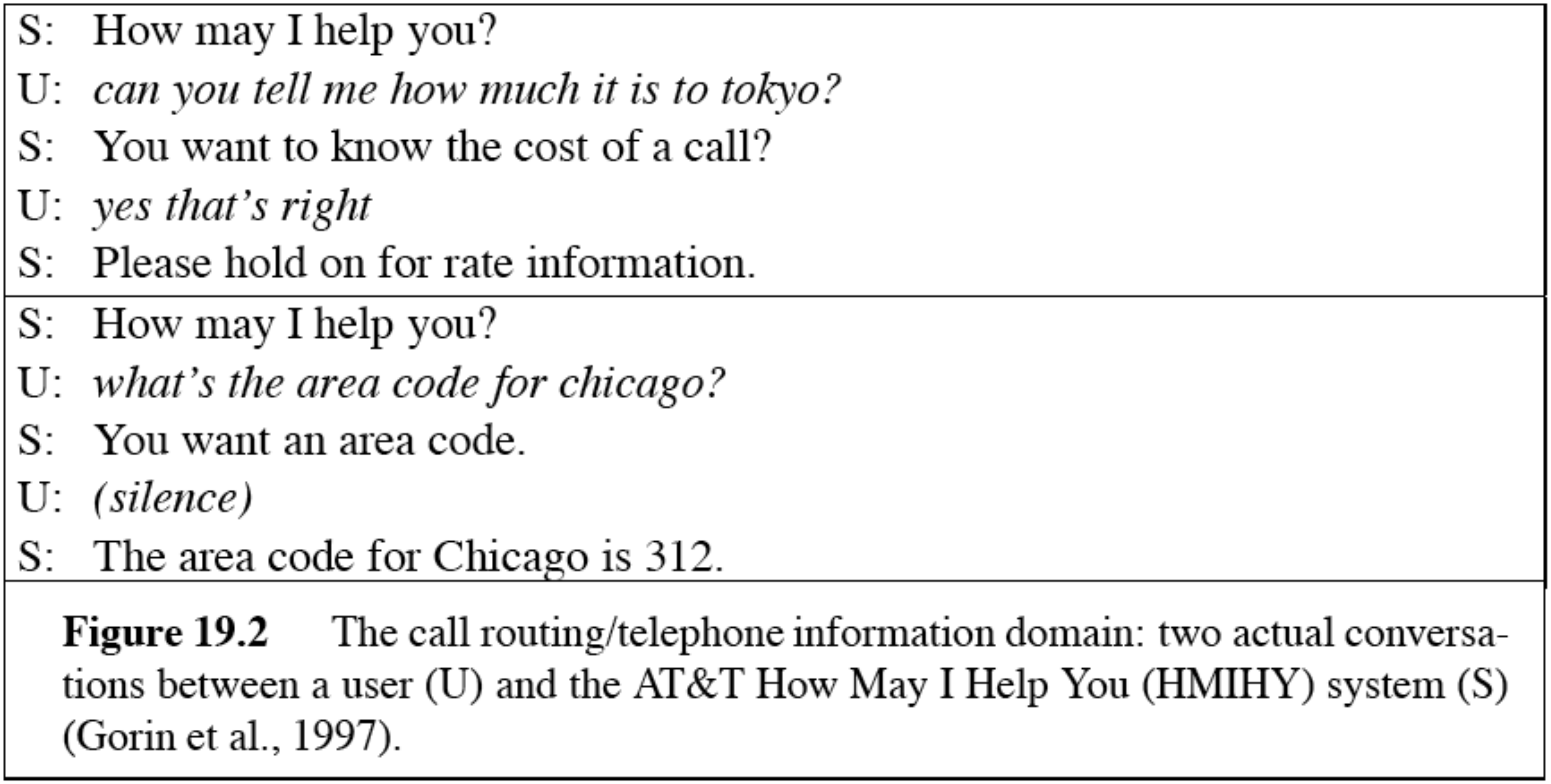
S14: a flight on wednesday.

S15: I have a flight departing Pittsburgh at ten forty five a.m., arrives Seattle at one fifty seven p.m. Is that OK?

U16: *Yes*

Figure 19.1 The travel domain: a fragment from a successful conversation between a user (U) and the Communicator system (S) of Xu and Rudnicky (2000).

Call routing: ATT HMIHY

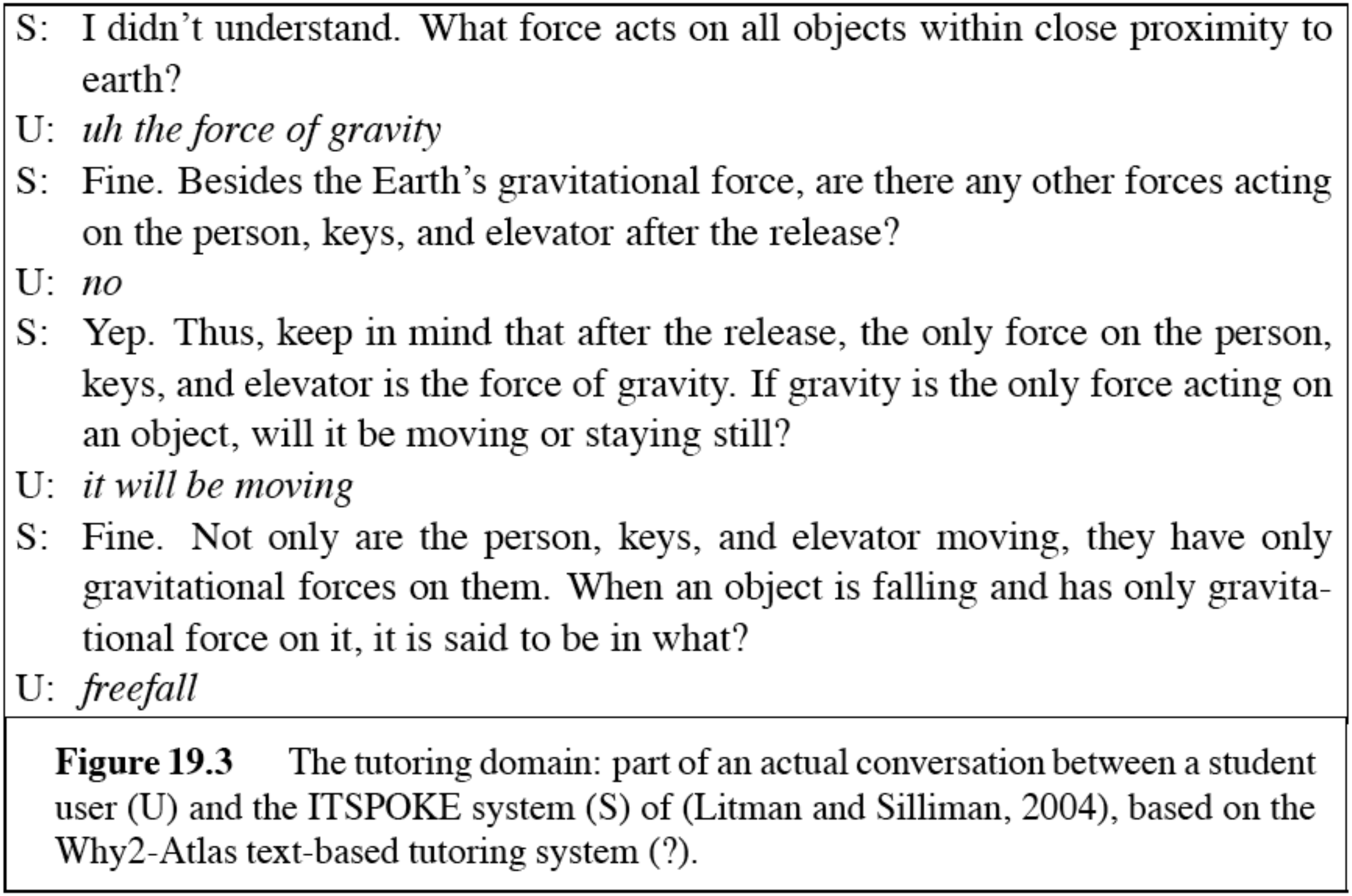


S: How may I help you?
U: *can you tell me how much it is to tokyo?*
S: You want to know the cost of a call?
U: *yes that's right*
S: Please hold on for rate information.

S: How may I help you?
U: *what's the area code for chicago?*
S: You want an area code.
U: *(silence)*
S: The area code for Chicago is 312.

Figure 19.2 The call routing/telephone information domain: two actual conversations between a user (U) and the AT&T How May I Help You (HMIHY) system (S) (Gorin et al., 1997).

A tutorial dialogue: ITSPOKE



S: I didn't understand. What force acts on all objects within close proximity to earth?

U: *uh the force of gravity*

S: Fine. Besides the Earth's gravitational force, are there any other forces acting on the person, keys, and elevator after the release?

U: *no*

S: Yep. Thus, keep in mind that after the release, the only force on the person, keys, and elevator is the force of gravity. If gravity is the only force acting on an object, will it be moving or staying still?

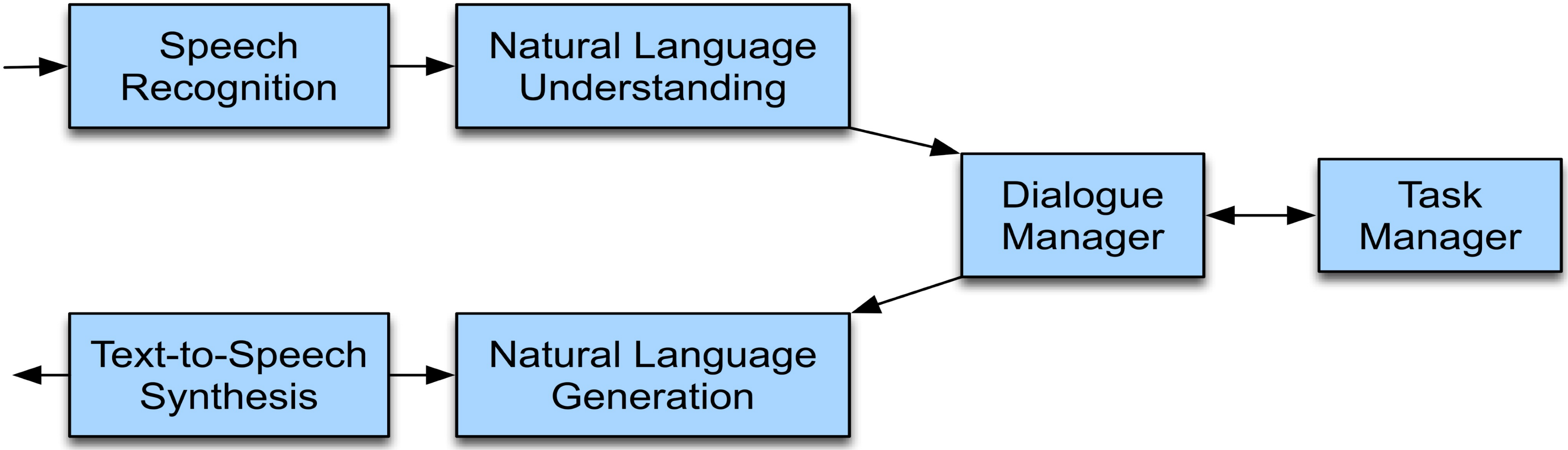
U: *it will be moving*

S: Fine. Not only are the person, keys, and elevator moving, they have only gravitational forces on them. When an object is falling and has only gravitational force on it, it is said to be in what?

U: *freefall*

Figure 19.3 The tutoring domain: part of an actual conversation between a student user (U) and the ITSPOKE system (S) of (Litman and Silliman, 2004), based on the Why2-Atlas text-based tutoring system (?).

Dialogue System Architecture



Dialogue Manager

Controls the architecture and structure of dialogue

- Takes input from ASR (speech recognizer) & NLU components
- Maintains some sort of internal state
- Interfaces with Task Manager
- Passes output to Natural Language Generation/ Text-to-speech modules

Four architectures for dialogue management

Finite State

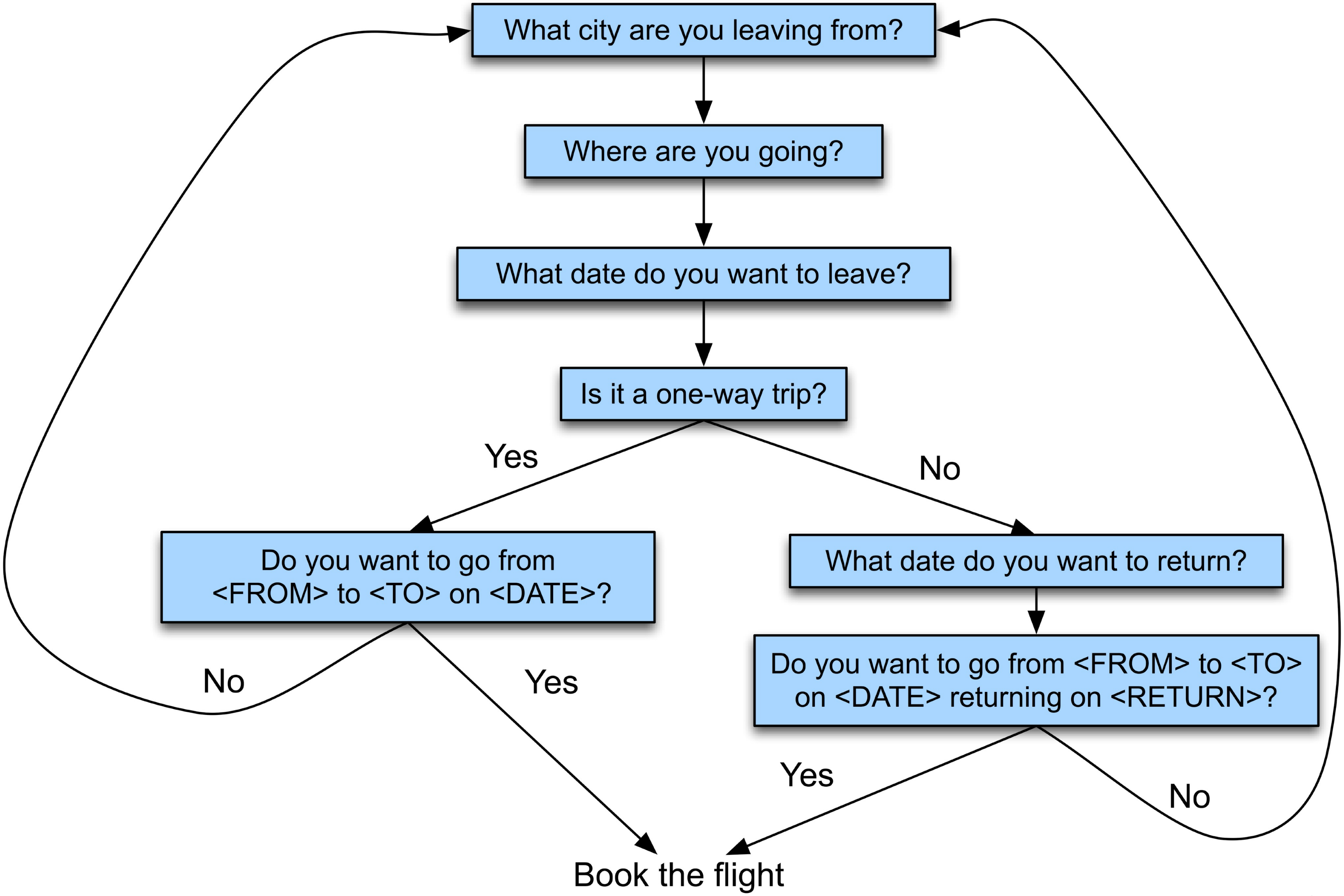
Frame-based

Information State

- Markov Decision Processes

AI Planning

Finite State Dialogue Manager


What city are you leaving from?
Where are you going?
What date do you want to leave?
Is it a one-way trip?
Yes
No
Do you want to go from <FROM> to <TO> on <DATE>?
What date do you want to return?
No
Yes
Do you want to go from <FROM> to <TO> on <DATE> returning on <RETURN>?
Yes
No
Book the flight

Finite-state dialogue managers

System completely controls the conversation with the user:

- It asks the user a series of questions
- It may ignore (or misinterpret) anything the user says that is not a direct answer to the system's questions

Systems that control conversation like this are system initiative or single initiative.

"Initiative": who has control of conversation

In normal human-human dialogue, initiative shifts back and forth between participants.

Task-driven dialog as slot filling

If the purpose of the dialog is to complete a specific task (e.g. to book a plane ticket), that task can often be represented as a **frame** with a number of slots to fill.

The task is completed if all the necessary slots are filled.

Frame-based dialog agents

Based on a "domain ontology"

A knowledge structure representing user intentions

One or more **frames**

Each a collection of **slots**

Each slot having a **value**

NLU with frame/slot semantics

There are many ways to represent the meaning of sentences. For speech dialogue systems, most common is “Frame slot semantics”:

Show me morning flights from Boston to SF on Tuesday.

```
SHOW:
   FLIGHTS:
         ORIGIN:
           CITY: Boston
           DATE:  Tuesday
           TIME:  morning
         DEST:
           CITY: San Francisco
```

The Frame

A set of **slots**, to be filled with information of a given **type**
Each associated with a **question** to the user

Slot	Type	Question
ORIGIN	city	What city are you leaving from?
DEST	city	Where are you going?
DEP DATE	date	What day would you like to leave?
DEP TIME	time	What time would you like to leave?

Information-State and Dialogue Acts

If we want a dialogue system to be more than just form-filling, it needs to be able to:

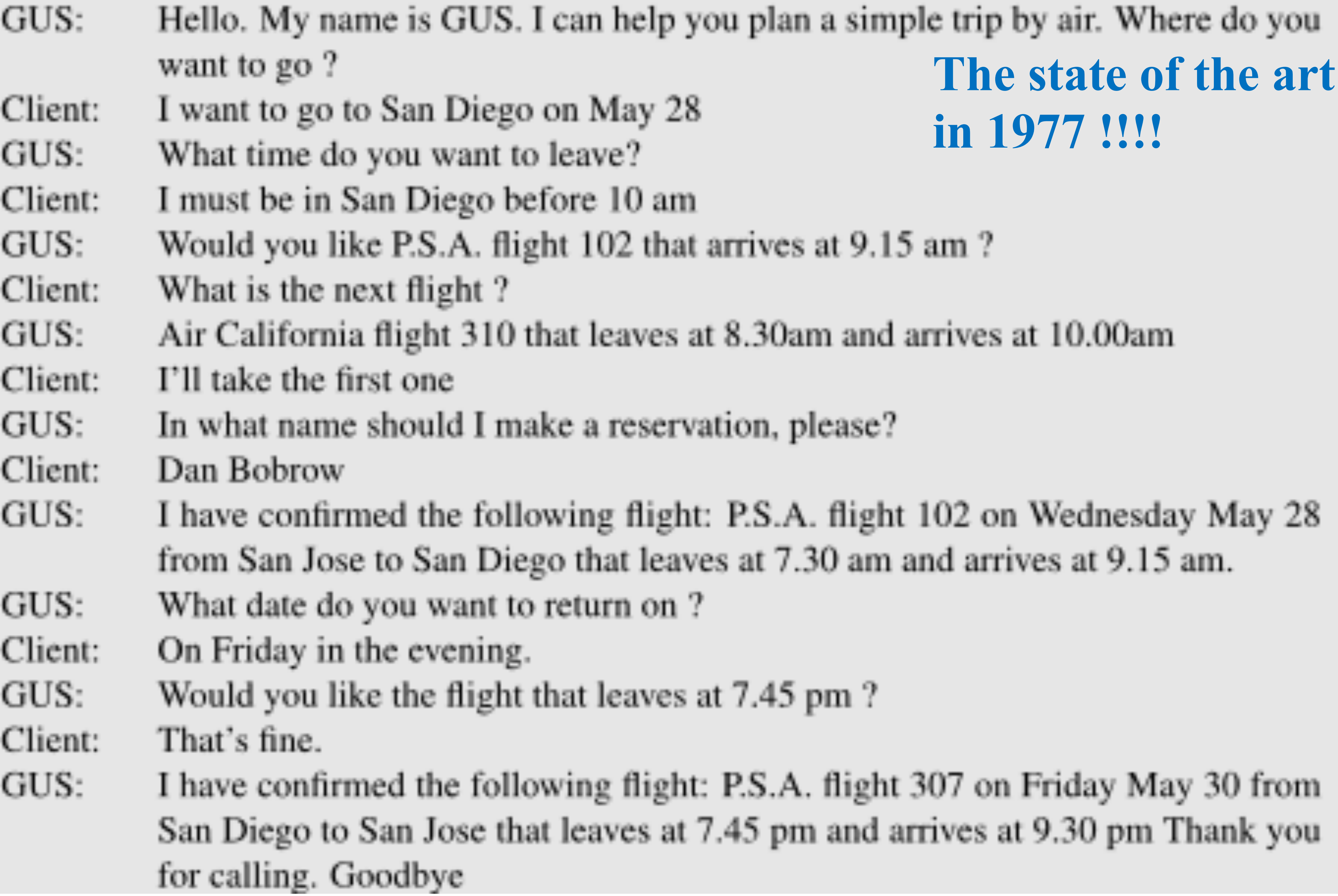
Decide when the user has asked a question, made a proposal, rejected a suggestion

Ground a user's utterance, ask clarification questions, suggestion plans

This suggests that:

Conversational agent needs sophisticated models of interpretation and generation

- In terms of speech acts and grounding
- Needs more sophisticated representation of dialogue context than just a list of slots



GUS: Hello. My name is GUS. I can help you plan a simple trip by air. Where do you want to go ?

Client: I want to go to San Diego on May 28

GUS: What time do you want to leave?

Client: I must be in San Diego before 10 am

GUS: Would you like P.S.A. flight 102 that arrives at 9.15 am ?

Client: What is the next flight ?

GUS: Air California flight 310 that leaves at 8.30am and arrives at 10.00am

Client: I'll take the first one

GUS: In what name should I make a reservation, please?

Client: Dan Bobrow

GUS: I have confirmed the following flight: P.S.A. flight 102 on Wednesday May 28 from San Jose to San Diego that leaves at 7.30 am and arrives at 9.15 am.

GUS: What date do you want to return on ?

Client: On Friday in the evening.

GUS: Would you like the flight that leaves at 7.45 pm ?

Client: That's fine.

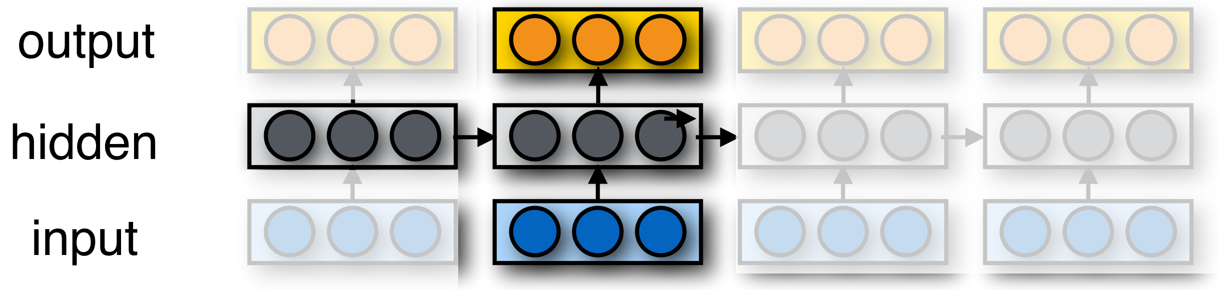
GUS: I have confirmed the following flight: P.S.A. flight 307 on Friday May 30 from San Diego to San Jose that leaves at 7.45 pm and arrives at 9.30 pm Thank you for calling. Goodbye

The state of the art in 1977 !!!!

Back to Neural Nets…

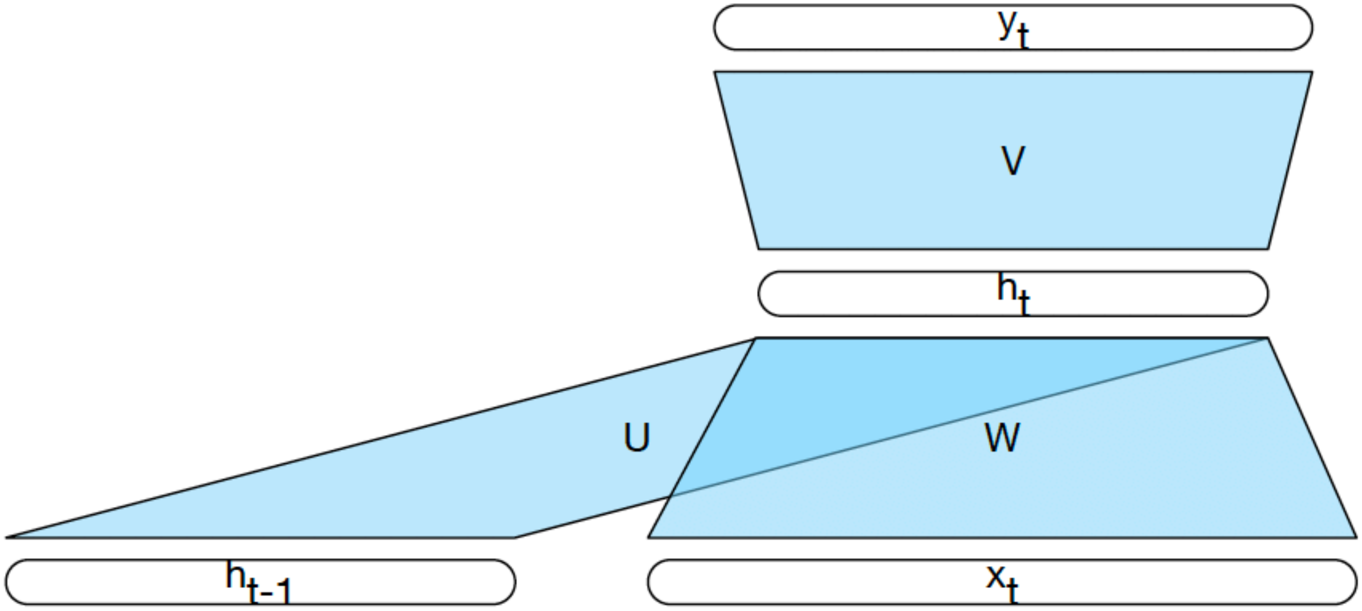
Basic RNNs

Each time step corresponds to a feedforward net where the hidden layer gets its input not just from the layer below but also from the activations of the hidden layer at the previous time step

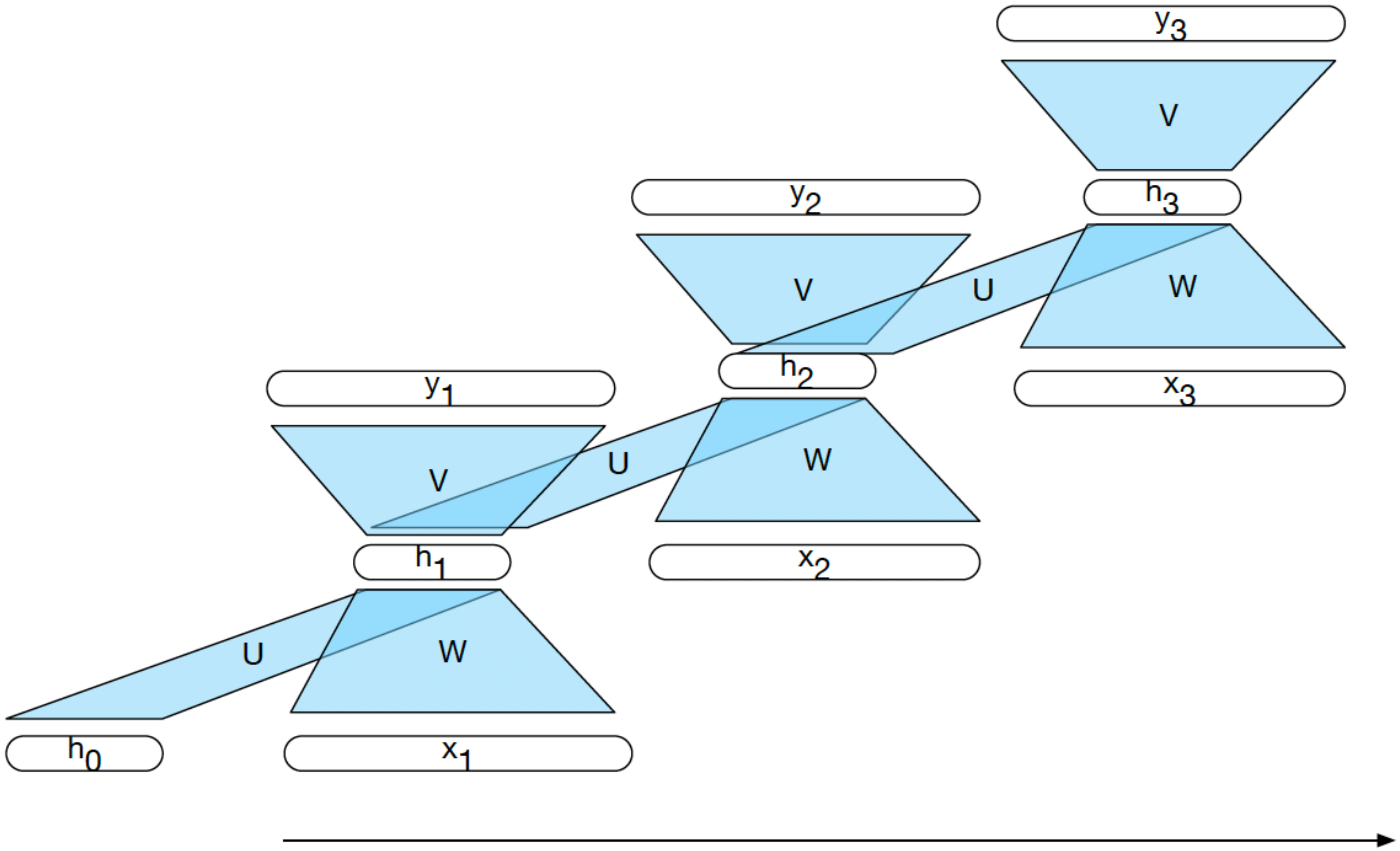


Basic RNNs

Each time step corresponds to a feedforward net where the hidden layer gets its input not just from the layer below but also from the activations of the hidden layer at the previous time step



A basic RNN unrolled in time

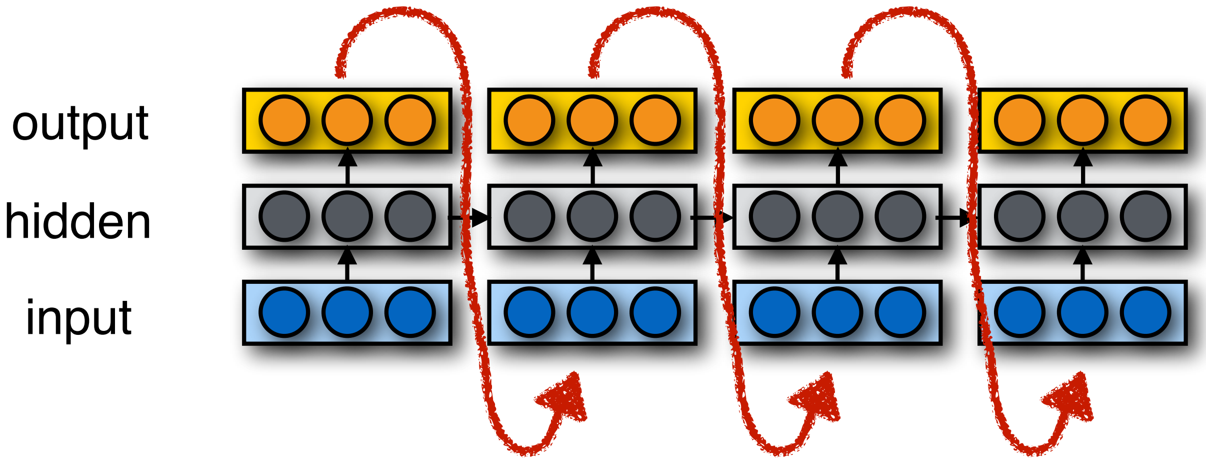


RNNs for generation

To generate a string $w_0 w_1 \ldots w_n\ w_{n+1}$ (where $w_0 = \text{<s>}$, and $w_{n+1} = \text{<\s>}$), give w_0 as first input, and then pick the next word according to the computed probability

$$P(w_i \mid w_0 \ldots w_{i-1})$$

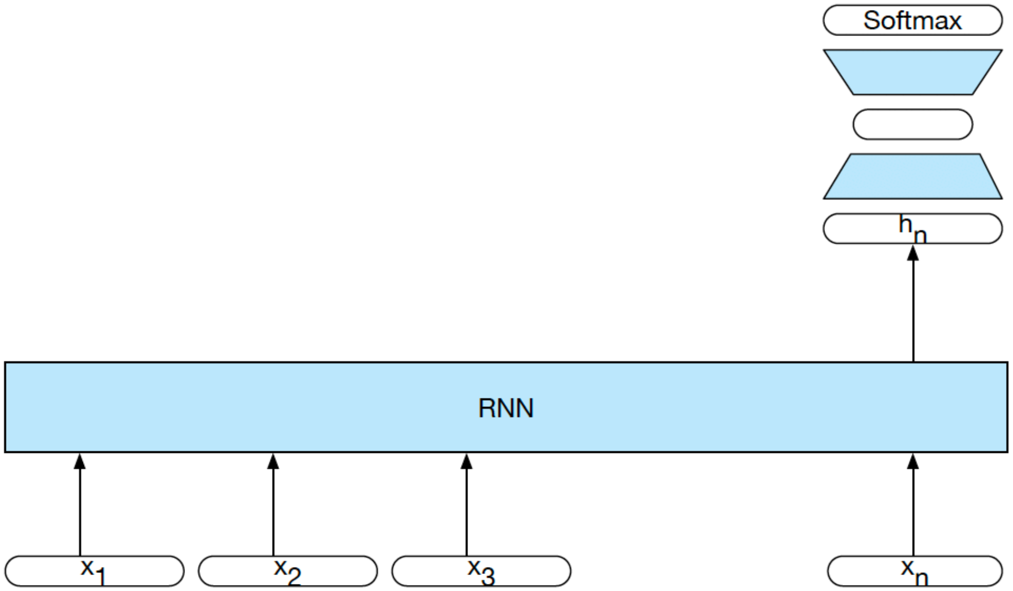
Feed this word in as input into the next layer.



RNNs for sequence classification

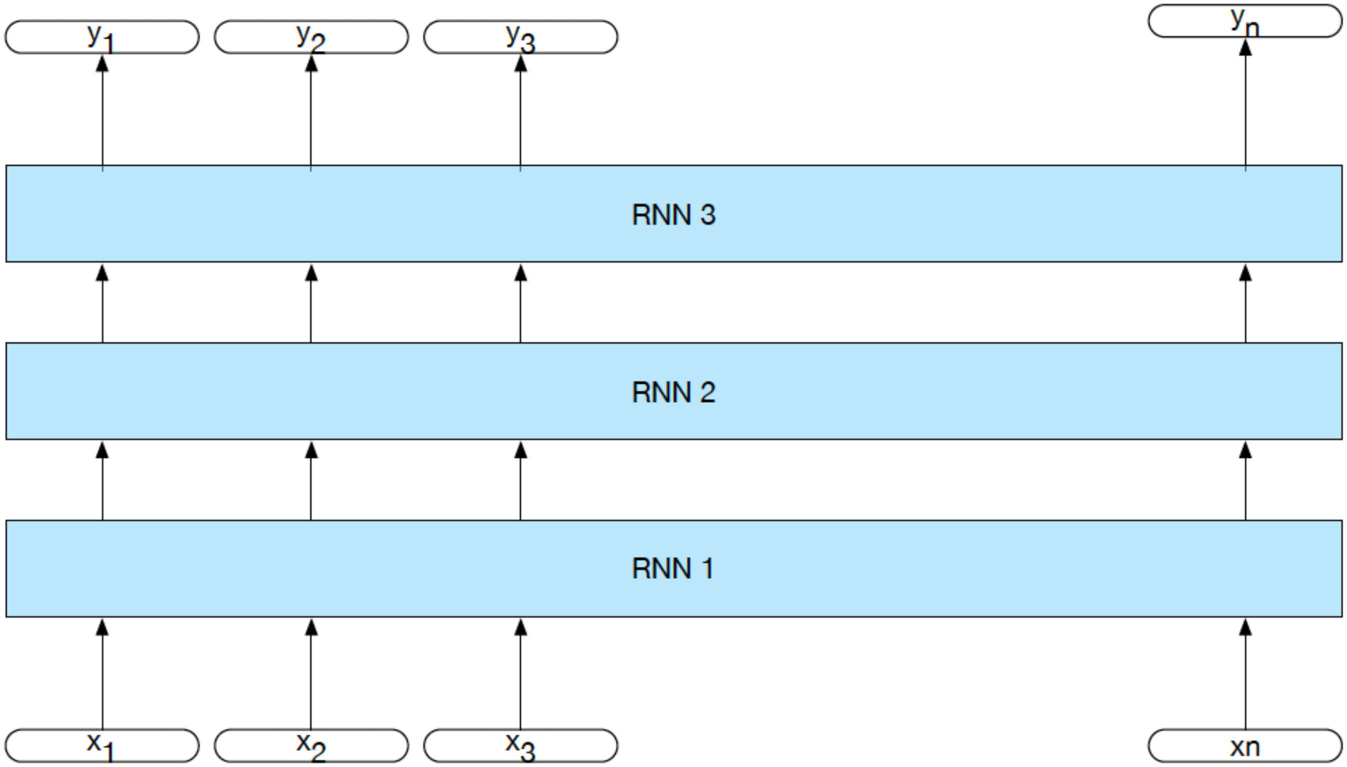
If we just want to assign a label to the entire sequence, we don't need to produce output at each time step, so we can use a simpler architecture.

We can use the hidden state of the last word in the sequence as input to a feedforward net:



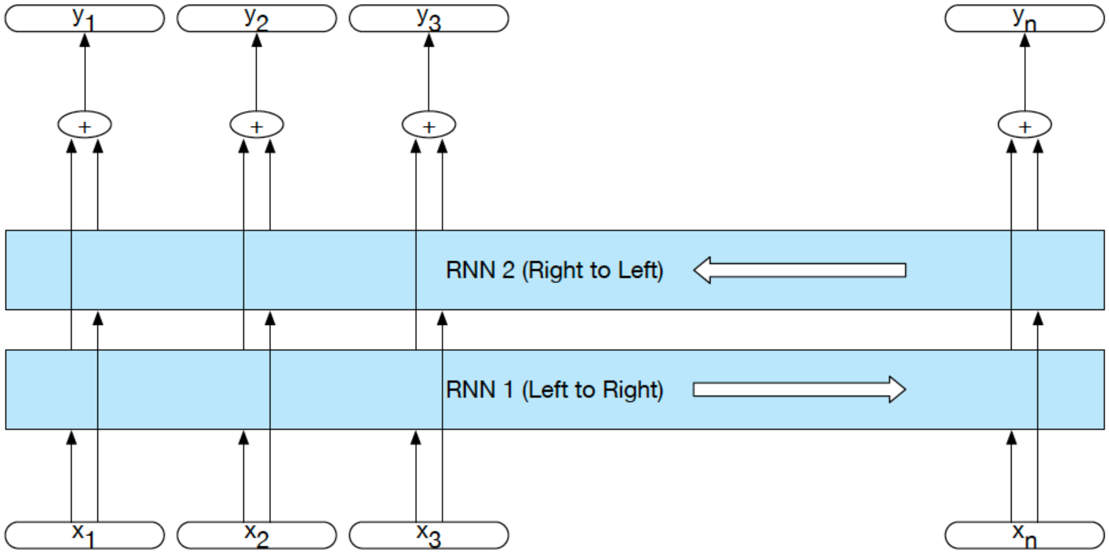
Stacked RNNs

We can create an RNN that has “vertical” depth (at each time step) by stacking:



Bidirectional RNNs

Unless we need to generate a sequence, we can run two RNNs over the input sequence — one in the forward direction, and one in the backward direction. Their hidden states will capture different context information.



Further extensions

Character and substring embeddings

We can also learn embeddings for individual letters.

This helps generalize better to rare words, typos, etc.

These embeddings can be combined with word embeddings (or used instead of an UNK embedding)

Context-dependent embeddings (ELMO, BERT, ….)

Word2Vec etc. are static embeddings: they induce a type-based lexicon that doesn't handle polysemy etc.

Context-dependent embeddings produce token-specific embeddings that depend on the particular context in which a word appears.

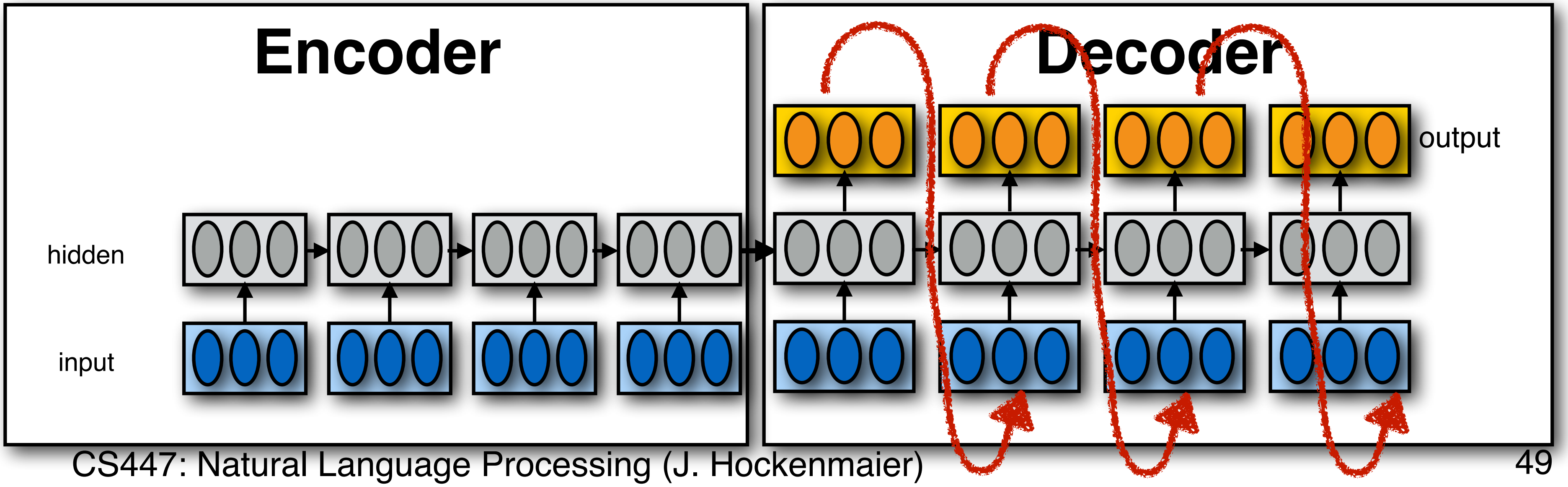
Encoder-Decoder Models (seq2seq)

Encoder-Decoder (seq2seq) model

Task: Read an input sequence and return an output sequence

- Machine translation: translate source into target language
- Dialog system/chatbot: generate a response

Reading the input sequence: RNN Encoder
Generating the output sequence: RNN Decoder



Encoder-Decoder (seq2seq) model

Encoder RNN:

reads in the input sequence

passes its last hidden state to the initial hidden state of the decoder

Decoder RNN:

generates the output sequence

typically uses different parameters from the encoder

may also use different input embeddings

Attention mechanisms

We want to condition the output generation of the decoder on a context-dependent representation of the input sequence.

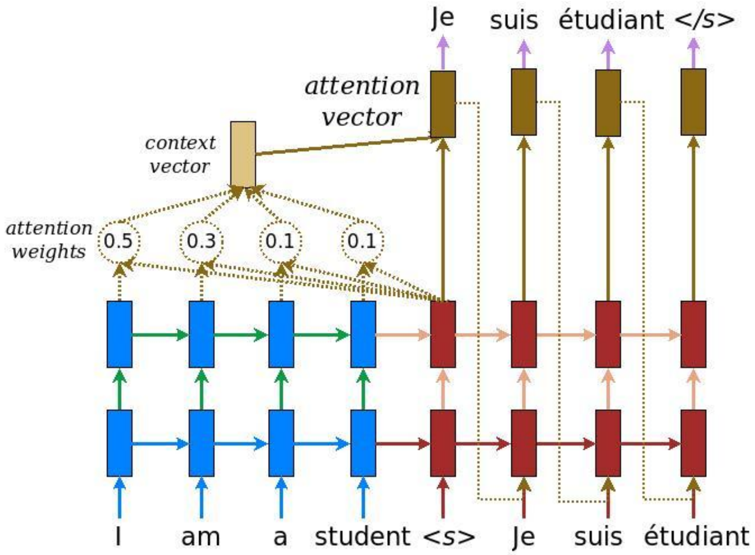
Attention computes a **distribution over the encoder's hidden states** (for the input sequence)

This distribution **depends on the decoder's hidden state** (and is computed anew for each output symbol)

The attention distribution is used to compute **a weighted average of the encoder's hidden state vectors**.

This **context-dependent embedding of the input sequence** is fed into the output of the decoder RNN.

Attention mechanisms



$$\alpha_{ts} = \frac{\exp\left(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'=1}^{S} \exp\left(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)} \quad \text{[Attention weights]} \quad (1)$$

$$\boldsymbol{c}_t = \sum_s \alpha_{ts} \bar{\boldsymbol{h}}_s \quad \text{[Context vector]} \quad (2)$$

$$\boldsymbol{a}_t = f(\boldsymbol{c}_t, \boldsymbol{h}_t) = \tanh(\boldsymbol{W}_{\boldsymbol{c}}[\boldsymbol{c}_t; \boldsymbol{h}_t]) \quad \text{[Attention vector]} \quad (3)$$

$$\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \boldsymbol{W} \bar{\boldsymbol{h}}_s & \text{[Luong's multiplicative style]} \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W}_{\boldsymbol{1}} \boldsymbol{h}_t + \boldsymbol{W}_{\boldsymbol{2}} \bar{\boldsymbol{h}}_s\right) & \text{[Bahdanau's additive style]} \end{cases} \quad (4)$$

https://colab.research.google.com/github/tensorflow/tensorflow/blob/master/tensorflow/contrib/eager/python/examples/nmt_with_attention/nmt_with_attention.ipynb#scrollTo=TNfHIF71ulLu