REINFORCEMENT LEARNING FOR DIALOGUE SYSTEMS

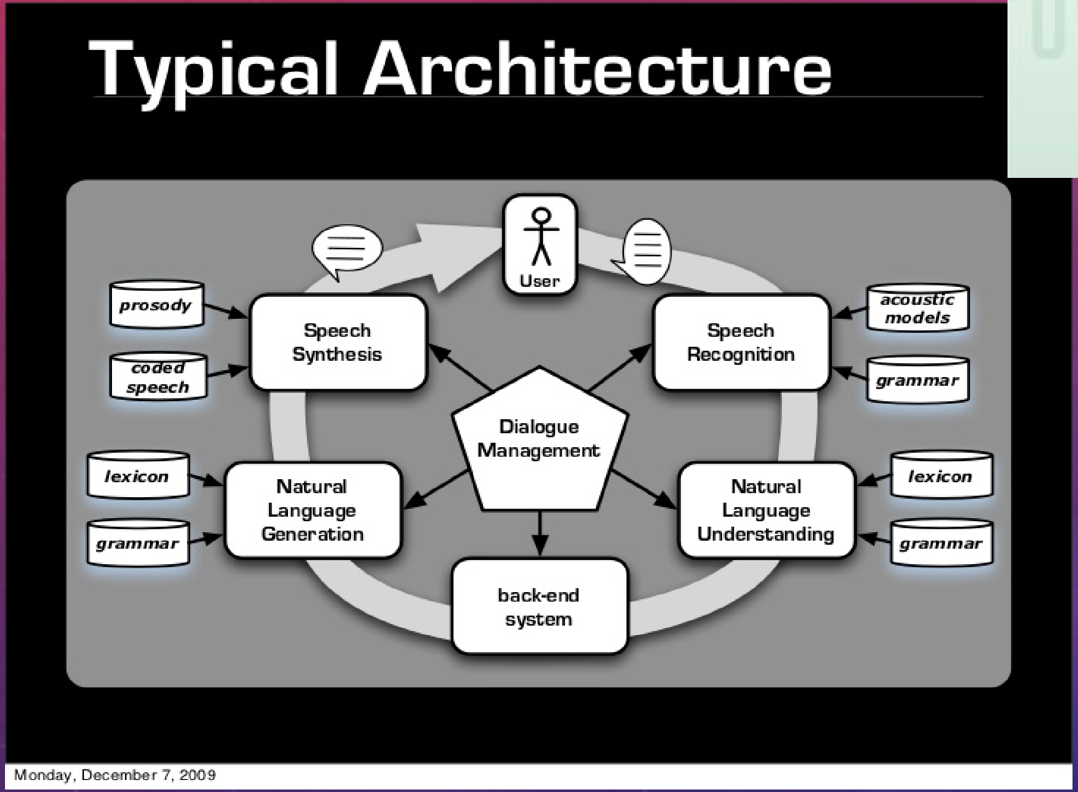
CHAPTER 1&2

MA LIN

EXCHANGE STUDENT

CONTENTS

INTRODUCTION



THE DESIGN PROBLEM FOR SPOKEN DIALOGUE SYSTEMS(SDS)

Dialogue strategy design is a difficult task

The design of Spoken Dialogue Systems(SDS) is not only concerned with integrating speech and language processing modules such as Automatic Speech Recognition(ASR), Spoken Language Understanding(SLU), Natural Language Generation(NLG), and Text-to-speech(TTS) synthesis systems, but also requires the development of skills for " What to say next" dialogue strategies:

- User's tasks, information-seeking, tutoring, user' preference or behavior

→ Great variability and unpredictability

Problem of conventional design

- Rule-based: re-design is necessary in order to produce good strategies
- Not re-usable: hand-coded strategies is not reusable from task to task
- Not-scalable: require amount of human labour and expertise

KEY POTENTIAL ADVANTAGES

- A data-driven automatic development cycle
- Provably optimal action policies
- A principled mathematical model for action selection
- Possibilities for generalisation to unseen states
- Reduced development and deployment costs.

Chapter 2

- First attempts to produce human speech - in the second half of the 18th century
  - One of the best known Wolfgang von Kempelen's speaking machine (*figure 1*)
  - The first that produced not only some speech sound but also whole words and short sentences

- Human-machine dialogue is far from resembling the capabilities of human-human dialogue

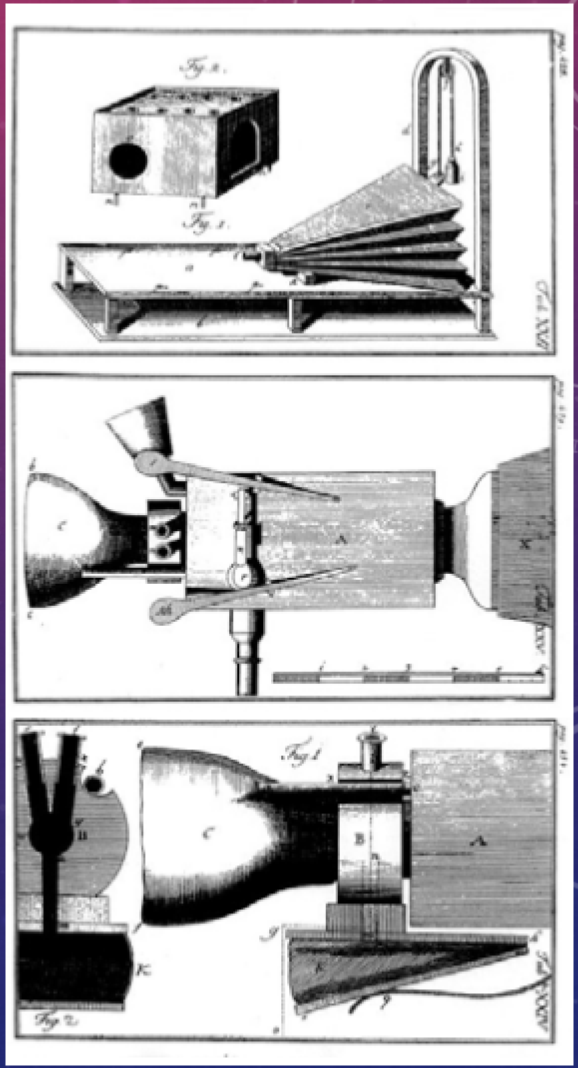
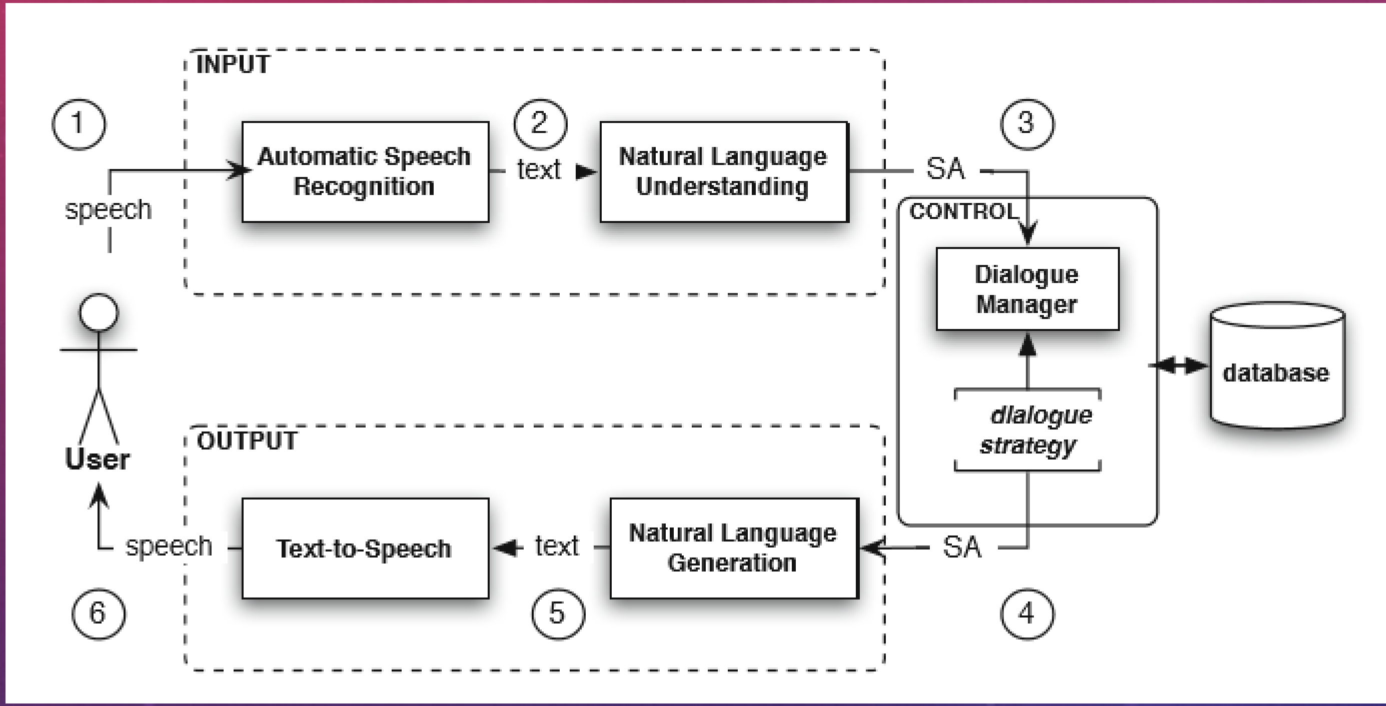


Figure 1

HUMAN-COMPUTER INTERACTION

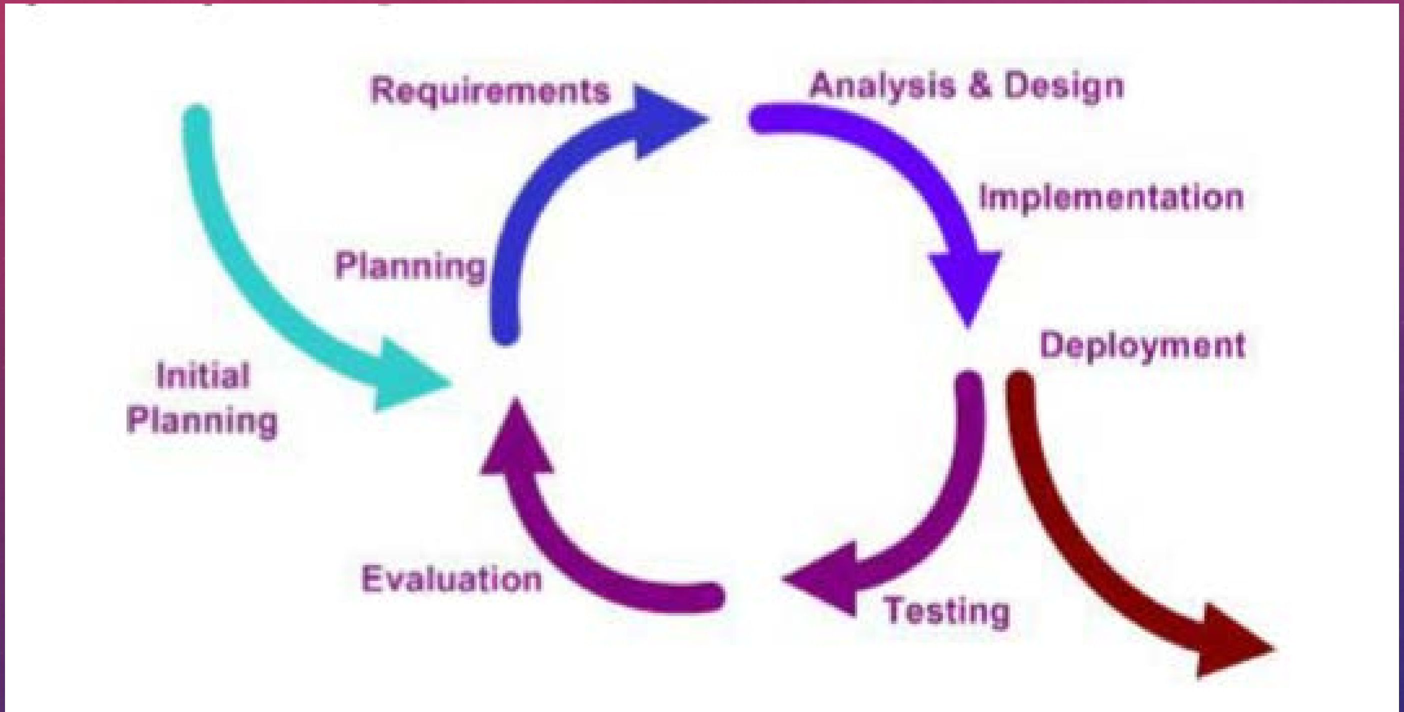


Dialogue system

- Human-Computer Interaction is the study of interaction between people(user) and computers(such as dialogue system)

- Differ from human-human dialogue:
  - lack of communicative skill (due to limited capabilities)
  - lack of language understanding (due to less-than-perfect input processing(ASR & NLU) and common use of shallow semantic representations)

CONVENTIONAL METHODS FOR DIALOGUE STRATEGY DEVELOPMENT



Conventional Development Lifecycle

1. Requirement analysis:
System designer examines the use case of the system and the language requirements

2. Initial strategy is designed:
Flow chat describe all the possible choice points for dialogue tasks and sub-tasks as a finite state automaton.

3. Implemented into a working dialogue system (VoiceXML)
Translating the design decisions into code.

4. Test and Evaluation

STRATEGY IN INDUSTRY AND RESERACH

There is a wide range of techniques to develop dialogue strategies, and technologies applied in industry are very different from the ones applied in research.

Quality Control in Industry:

Criterion is defined by Return-On-Investment (Ratio of money gained or lost on an investment relative the amount of money invested)

Evaluation Practices in Academia :

- PARADISE (PARAdigm for Dialogue System Evaluation)
- SASSI (Subjective Assessment of Speech System Interfaces)

STRATEGY IMPLEMENTATION

- **Implementation Practices in Industry**

Most commercial systems rely on Finite State Automata(FSA) controlled by menus, forms, or frames. The most common applications are form filling dialogues, information retrieval, transactions and services

- **Implementation Practices in Academia**

Most research systems to date have been based either on planning with logical inference, or they are implemented in the “Information State Update” (ISU) approach using frames or tree sub-structures as control mechanism

MACHINE LEARNING PARADIGMS

Definition of Machine Learning :
Given a specific task to solve, and a class of functions F, learning means using a set of observations, in order to find $f* \in F$ which solves the task in an optimal sense. This entails defining a cost function C: $F \rightarrow \Re$ such that, for the optimal solution $\forall f \in F$, $f*$, $C(f*) \leq C(f)$ (no solution has a cost less than the cost of the optimal solution).

- **Supervise Learning**
- **Unsupervised Learning**
- **Reinforcement Learning**

MACHINE LEARNING PARADIGMS -SL

- **Supervise Learning**

In Supervised Learning (SL), we are given a set of example pairs/labelled data points (x,y), $x \in X$, $y \in Y$ and the aim is to find a function f in the allowed class of functions that matches the examples. In other words, we wish to infer the mapping implied by the data; the cost function is to reduce the mismatch between our mapping and the data. The goal is to find a model which mimics the data as close as possible, while still being general enough to classify/predict unseen events well.

- **Example of Application of SL in dialogue system**
- Adapts dialogue strategies to various user and situation models via example-based learning. The training corpus is gathered using the following setup: a set of possible system responsesisdisplayedonascreenwhiletheuserinteractswiththesystem.Foreach system turn, the user selects the response that they think is most suitable in the current situation. The learned strategy chooses the action which is selected most often by the users.

Short-coming : human-assisted (user are not experts), has high costs compared to strategy design by an expert

MACHINE LEARNING PARADIGMS

Similar to SL ,this approach is based on some local cost function, defining a mapping states and actions, which is here called utility .In addition, this approach also explicitly models the uncertainty in the observed state. In this framework the agent selects the action A = a that maximizes expected utility, EU(a|o), where o are observed events. Action selection is guided by the following optimisation:

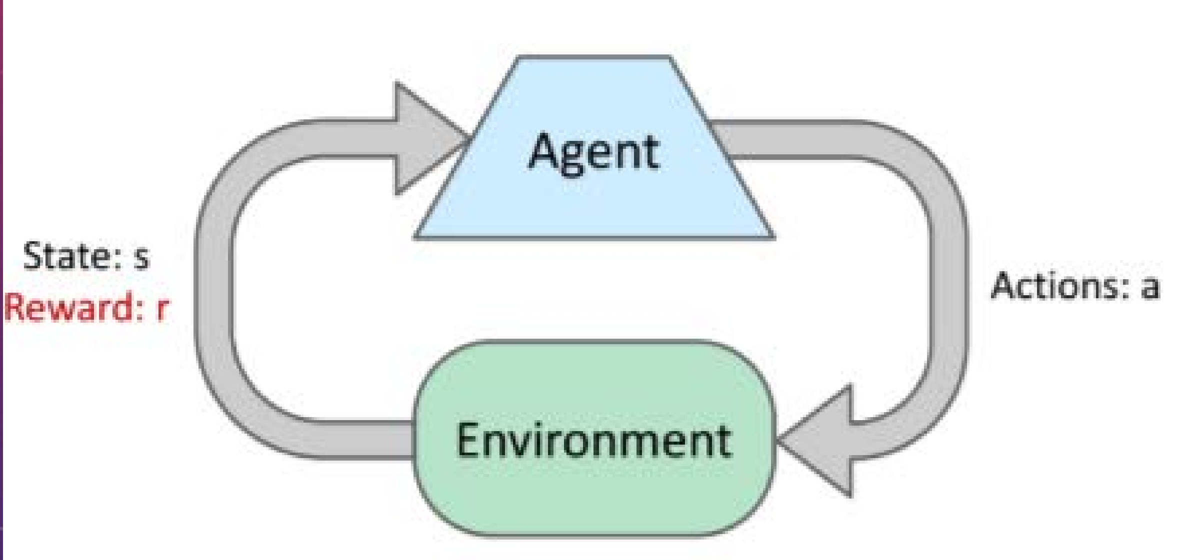
$$A = \text{argmax} EU(a|o) = \text{argmax} \sum s\, P(S = s|o) \times \text{utility}(a,s);$$

where utility(a,s) expresses the utility of taking action a when the state of the world is s. The utility function is trained via “local” user ratings.

Shortcoming : No consider what action is best in the long run

MACHINE LEARNING PARADIGMS -RL

Main idea : In contrast to the above approaches, Reinforcement Learning treats dialogue strategy learning as a sequential optimisation problem, leading to strategies which are globally optimal



- RL is a general-purpose framework for artificial intelligence
  - RL is for an agent with the capacity to act
  - Each action a_t, influences the agent's future state s_t
  - Success is measured by a *scalar* reward r_t
  - Must (learn to) act so as to maximize expected rewards

MACHINE LEARNING PARADIGMS -RL

- Markov Decision Processes: uncertainty can be explicitly represented in RL. Stochastic variation in the user response is represented as transition probabilities between states and actions using MDP
- Urgent problems for RL-based strategy development: RL need substantial amounts of data to learn reliable strategies
- HOW? Agent has to GUESS

SUMMARY