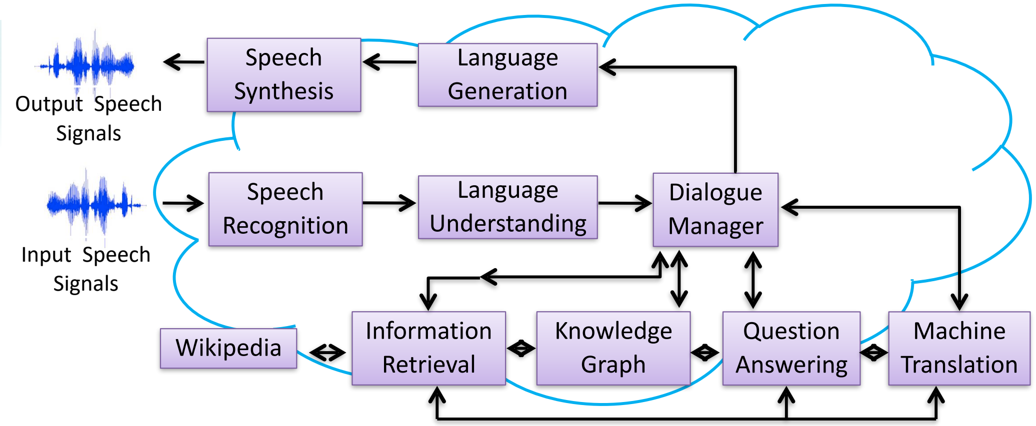
17.0 Spoken Dialogues

References: 1. 11.1 - 11.2.1, Chapter 17 of Huang

2. "Conversational Interfaces: Advances and Challenges", Proceedings of the IEEE, Aug 2000
3. "The AT&T spoken language understanding system", IEEE Trans. on Speech and Audio Processing, vol.14, no.1, pp.213-222, 2006
4. "Talking to machine" in ICSLP, 2002
5. "A telephone-based conversational interface for weather information" IEEE Trans. On Speech and Audio Processing, vol. 8, no. 1, pp. 85-96, 2000.
6. "Spoken Language Understanding", IEEE Signal Processing Magazine, vol.22, no. 5, pp. 16-31, 2005
7. "Spoken Language Understanding", IEEE Signal Processing Magazine, May 2008

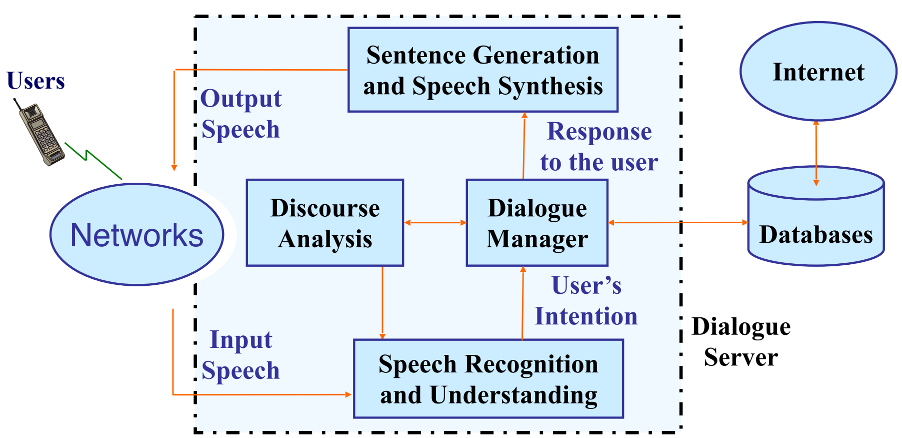
Well-Known Application Examples of Speech and Language Technologies – Speaking Personal Assistant

- **Examples**
  – Weather in New York next week ?
  – Who is the president of US ? What did he say today ?
  – How can I go to National Taiwan University ?
  – Short messaging, personal scheduling, etc.
- **Special Questions:**
  – 唐詩宋詞, 出師表…
  – 說個笑話…



Spoken Dialogue Systems

- **Almost all human-network interactions can be made by spoken dialogue**
- **Speech understanding, speech synthesis, dialogue management, discourse analysis**
- **System/user/mixed initiatives**
- **Reliability/efficiency, dialogue modeling/flow control**
- **Transaction success rate/average dialogue turns**



Key Processes in A Spoken Dialogue

- **A Basic Formulation**

$$A_n^* = \arg\max_{A_n} \text{Prob}(A_n|X_n, S_{n-1})$$

X_n: speech input from the user in the n-th dialogue turn
S_n: discourse semantics (dialogue state) at the n-th dialogue turn
A_n: action (response, actions, etc.) of the system (computer, hand-held device, network server, etc.) after the n-th dialogue turn

– goal: the system takes the right actions after each dialogue turn and complete the task successfully finally

$$A_n^* \approx \arg\max_{A_n, S_n} \underbrace{P(A_n|S_n)}_{\text{by dialogue management}} \sum_{F_n} \underbrace{P(S_n|F_n, S_{n-1})}_{\text{by discourse analysis}} \underbrace{P(F_n|X_n, S_{n-1})}_{\text{by speech recognition and understanding}}$$

F_n: semantic interpretation of the input speech X_n

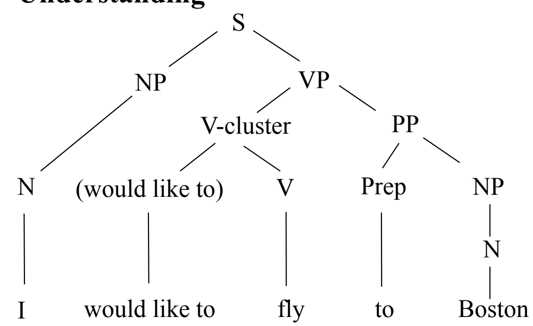
- **Three Key Elements**
  – speech recognition and understanding: converting X_n to some semantic interpretation F_n
  – discourse analysis: converting S_{n-1} to S_n, the new discourse semantics (dialogue state), given all possible F_n
  – dialogue management: select the most suitable action A_n given the discourse semantics (dialogue state) S_n

Dialogue Structure

- **Turns**
  - an uninterrupted stream of speech(one or several utterances/sentences) from one participant in a dialogue
  - speaking turn: conveys new information
    back-channel turn: acknowledgement and so on(e.g. O. K.)
- **Initiative-Response Pair**
  - a turn may include both a response and an initiative
  - system initiative: the system always leads the interaction flow
    user initiative: the user decides how to proceed
    mixed initiative: both acceptable to some degree
- **Speech Acts(Dialogue Acts)**
  - goal or intention carried by the speech regardless of the detailed linguistic form
  - forward looking acts
    - conversation opening(e.g. May I help you?), offer(e.g. There are three flights to Taipei…), assert(e.g. I'll leave on Tuesday), reassert(e.g. No, I said Tuesday), information request(e.g. When does it depart?), etc.
  - backward looking acts
    - accept(e.g. Yes), accept-part(e.g. O.K., but economy class), reject(e.g. No), signal not clear(e.g. What did you say?), etc.
  - speech acts ⇔ linguistic forms : a many-to-many mapping
    - e.g. "O.K." — request for confirmation, confirmation
  - task dependent/independent
  - helpful in analysis, modeling, training, system design, etc.
- **Sub-dialogues**
  - e.g. "asking for destination", "asking for departure time", …..

Language Understanding for Limited Domain

- **Semantic Frames — An Example for Semantic Representation**
  - a semantic class defined by an entity and a number of attributes(or slots)
    e.g. [Flight]:
    [Airline] → (United)
    [Origin] → (San Francisco)
    [Destination] → (Boston)
    [Date] → (May 18)
    [Flight No] → (2306)
  - "slot-and-filler" structure
- **Sentence Parsing with Context-free Grammar (CFG) for Language Understanding**



Grammar(Rewrite Rules)
S → NP VP
NP → N
VP → V-cluster PP
V-cluster → (would like to) V
V → fly| go
PP → Prep NP
N → Boston | I
Prep → to

  - extension to Probabilistic CFG, integration with N-gram(local relation without semantics), etc.

Robust Parsing for Speech Understanding

- **Problems for Sentence Parsing with CFG**
  - ungrammatical utterances
  - speech recognition errors (substitutions, deletions, insertions)
  - spontaneous speech problems: um–, cough, hesitation, repetition, repair, etc.
  - unnecessary details, irrelevant words, greetings, unlimited number of linguistic forms for a given act
    e.g. to Boston
    I'm going to Boston, I need be to at Boston Tomorrow
    um– just a minute– I wish to – I wish to – go to Boston
- **Robust Parsing as an Example Approach**
  - small grammars for particular items in a very limited domain, others handled as fillers
    e.g. Destination→ Prep CityName
    Prep → to |for| at
    CityName → Boston |Los Angeles|...
  - different small grammars may operate simultaneously
  - keyword spotting helpful
  - concept N-gram may be helpful

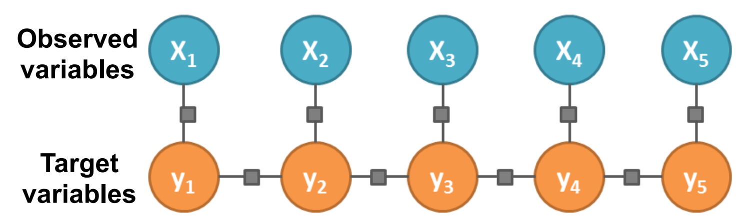
    Prob(c_i|c_{i-1}), c_i: concept
    CityName — direction (to, for...)
    (Boston,...)similar to class-based N-gram
- **Speech Understanding**
  - two-stage: speech recognition (or keyword spotting) followed by semantic parsing (e.g. robust parsing)
  - single-stage: integrated into a single stage

Conditional Random Field (CRF)

- **Find a label sequence *y* that maximizes:**

$$p(\boldsymbol{y}\,|\,\boldsymbol{x};\theta) = \frac{1}{Z(x)}\exp\{\sum_{i=1}^{M}\theta \bullet f(y_{i-1}, y_i, x_i)\}$$

  - Input observation sequence $\boldsymbol{x} = (x_1,\ x_2, \ldots,\ x_M)$
  - Output label sequence $\boldsymbol{y} = (y_1, y_2, \ldots, y_M)$
  - $f(y_{i-1}, y_i, x_i)$: feature function vector
  - θ: weights
  - $Z(x)$: term for normalization



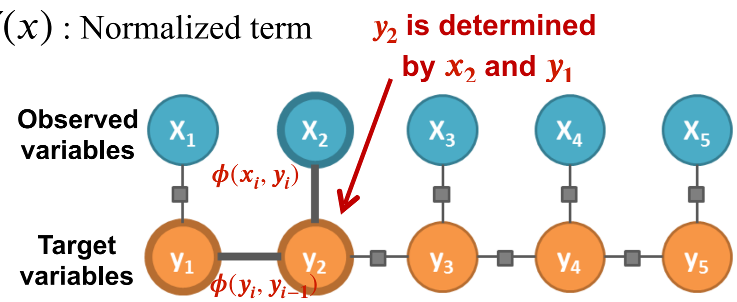
Conditional Random Field (CRF)

- **Find a label sequence y that maximizes:**

$$p(\boldsymbol{y} \mid \boldsymbol{x};\theta) = \frac{1}{Z(x)} \exp\{ \sum_{i=1}^{M} \theta \bullet f(y_{i-1}, y_i, x_i) \}$$

$\phi(x_i, y_i)\phi(y_i, y_{i-1})$

  - Input observation sequence $\boldsymbol{x} = (x_1, x_2, \ldots, x_M)$
  - Output label sequence $\boldsymbol{y} = (y_1, y_2, \ldots, y_M)$
  - $f(y_{i-1}, y_i, x_i)$: feature function vector
  - θ: weights
  - $Z(x)$: Normalized term

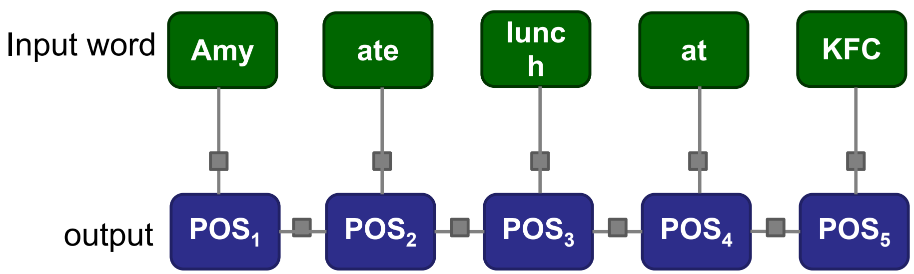


Example

- **POS Tagging**
  - Input sequence: natural language sentence
    - Ex: "Amy ate lunch at KFC"
  - Output sequence: POS tagging
    - Possible POS tagging: NOUN, VERB, ADJECTIVE, ADVERB, PREPOSITION…
    - Ex: "Amy(NOUN) ate(VERB) lunch(NOUN) at(PREPOSITION) KFC(NOUN)"

Example

- **POS Tagging**



  - POS_i is determined by the $word_i$ and POS_{i-1}

Training/Testing of CRF

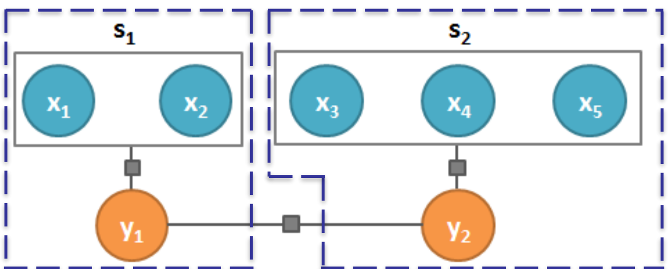
- **Training**
  - Find a parameter set θ to maximize the conditioned likelihood function $p(y \mid x;\theta)$ for the training set
  - Represent $p(y \mid x;\theta)$ as log likelihood function
    - $\log\left(p(\boldsymbol{y} \mid \boldsymbol{x};\theta)\right)$
    - solved by gradient descent algorithm
- **Testing**
  - Find a label sequence y that maximizes the conditioned likelihood function $p(y \mid x;\theta)$ for the input x
  - Solved by forward-backward and Viterbi algorithms

Semi-conditional Random Field (Semi-CRF)

- **Semi-CRF uses "phrase" instead of "word"**
- **To find the phrase and corresponding label sequence S that maximize:**

$$p(S|x) = \frac{1}{Z(x)}\exp\{\sum_{j=1}^{N} \theta \bullet f(y_{j-1}, y_j, \boldsymbol{x}, s_j)\}$$

  – Where s_j is a phrase in input sequence $\boldsymbol{x}$ and its label y_j
  – $S = \left(s_j, \ \ j = 1, 2, \ \cdots N\right)$
  – s_j is known in training but unknown in testing

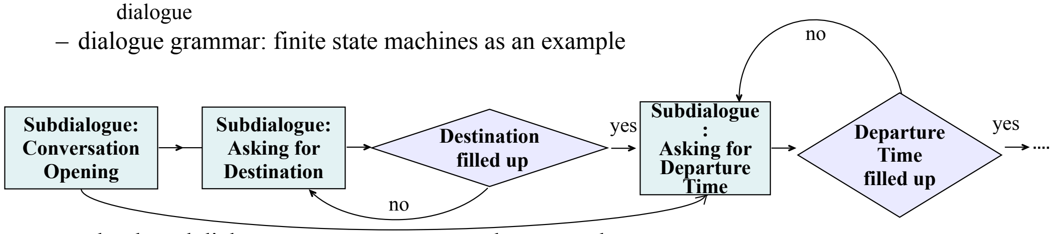


Example

- **Slot filling**
  – Input sequence: natural language sentence
    - Ex: Funny movie about bridesmaid starring Keira Knightley
  – Output sequence: slot sequence
    - GENRE, PLOT, ACTOR
    - Ex: [Funny](GENRE) movie about [bridesmaid](PLOT) starring [Keira Knightley](ACTOR)

Discourse Analysis and Dialogue Management

- **Discourse Analysis**
  – conversion from relative expressions(e.g. tomorrow, next week, he, it…) to real objects
  – automatic inference: deciding on missing information based on available knowledge(e.g. "how many flights in the morning? " implies the destination/origin previously mentioned)
  – inconsistency/ambiguity detection (e.g. need clarification by confirmation)
  – example approach: maintaining/updating the dialogue states(or semantic slots)
- **Dialogue Management**
  – controlling the dialogue flow, interacting with the user, generating the next action
    - e.g. asking for incomplete information, confirmation, clarify inconsistency, filling up the empty slots one-by-one towards the completion of the task, optimizing the accuracy/efficiency/user friendliness of the dialogue
  – dialogue grammar: finite state machines as an example

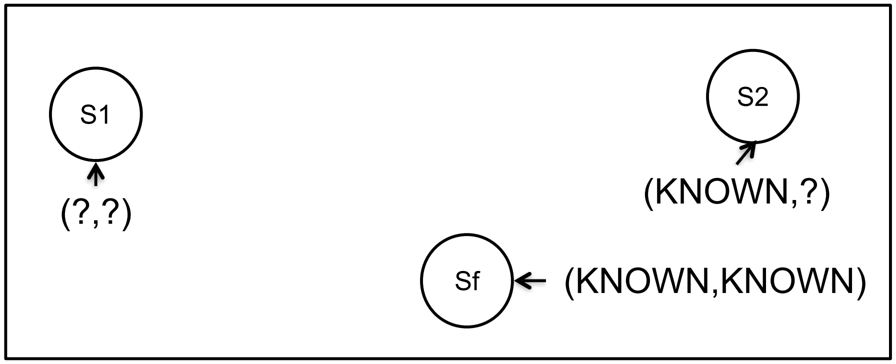


  – plan-based dialogue management as another example
  – challenging for mixed-initiative dialogues
- **Performance Measure**
  – internal: word error rate, slot accuracy (for understanding), etc.
  – overall: average success rate (for accuracy), average number of turns (for efficiency), etc.

Dialogue Management

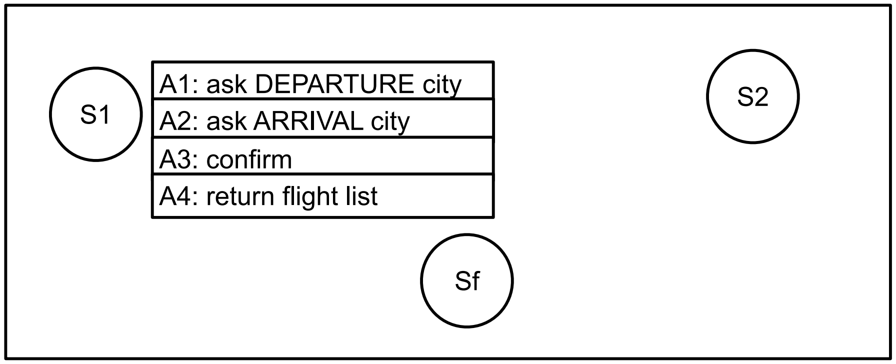
- **Example Approach – MDP-based**
- **Example Task: flight booking**
  – The information the system needs to know:
    - The departure city
    - The arrival city
  – Define the state as (DEPARTURE,ARRIVAL)
  – There are totally four states:
  – (?,?), (KNOWN,?), (?,KNOWN), (KNOWN,KNOWN)

Flight Booking with MDP (1/5)



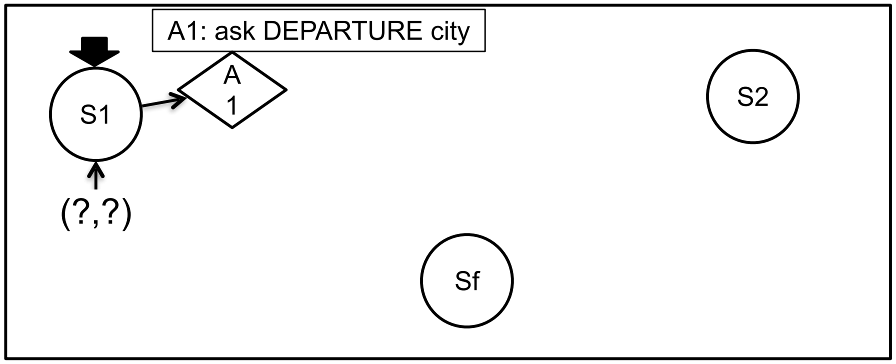
- The state is decided by the information the system knows.

Flight Booking with MDP (1/5)



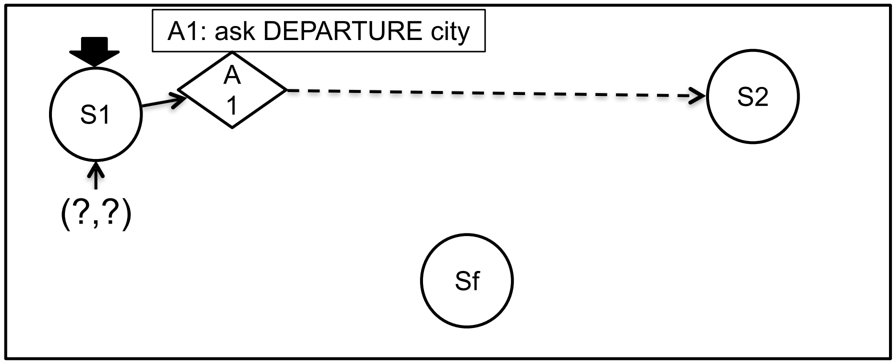
- The state is decided by the information the system knows.
- A set of available actions is also defined.

Flight Booking with MDP (2/5)



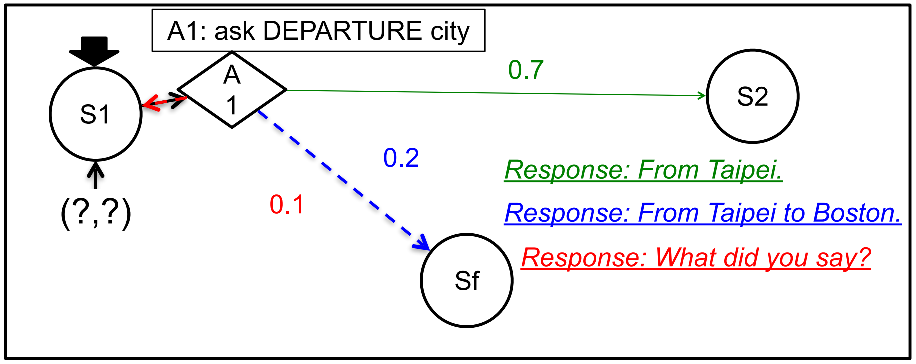
- Assume the system is at state S1 and takes action A1.

Flight Booking with MDP (2/5)



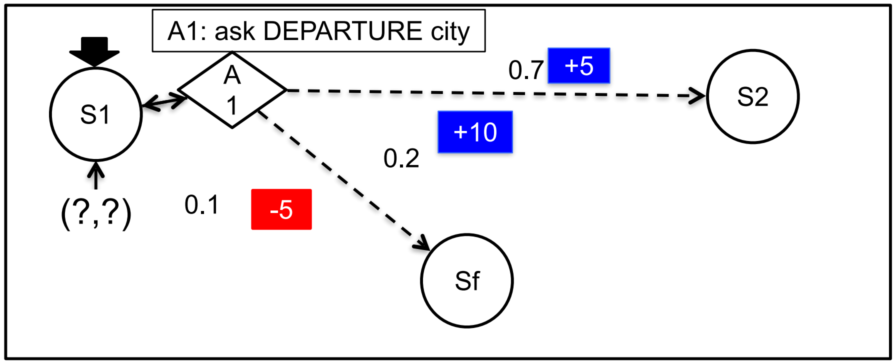
- Assume the system is at state S1 and takes action A1.
- User response will cause the state to transit.

Flight Booking with MDP (3/5)



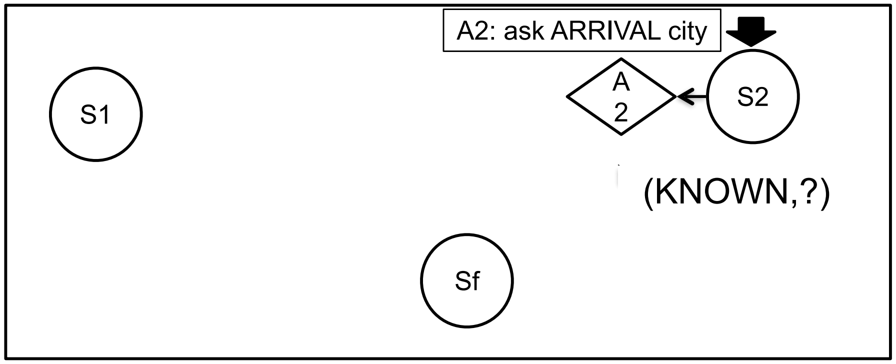
- **The transition is probabilistic based on user response and recognition results (with errors).**

Flight Booking with MDP (3/5)



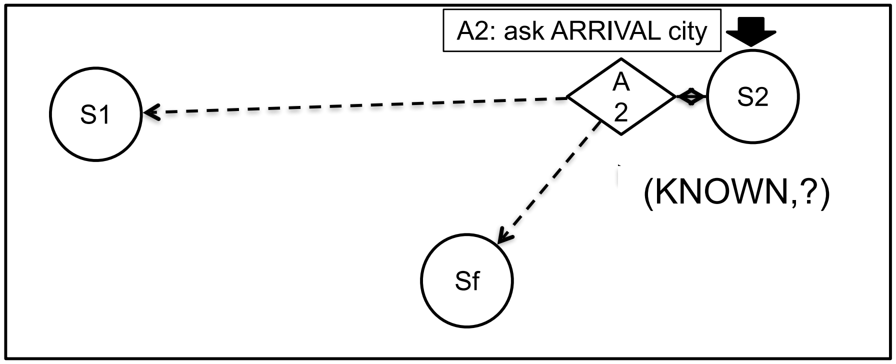
- **The transition is probabilistic based on user response and recognition results (with errors).**
- **A reward associated with each transition.**

Flight Booking with MDP (4/5)



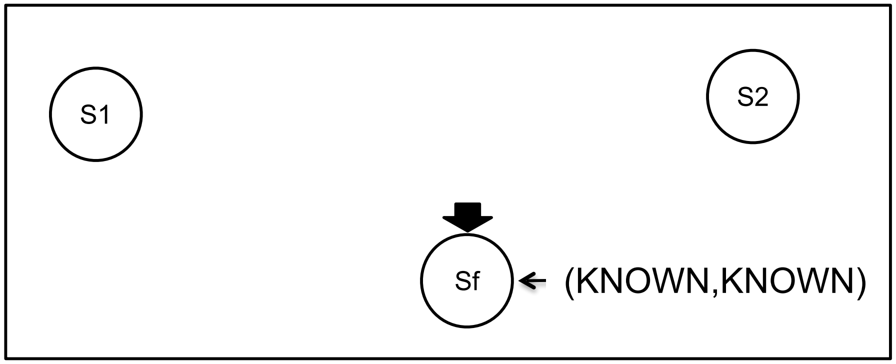
- **The interaction continues.**

Flight Booking with MDP (4/5)



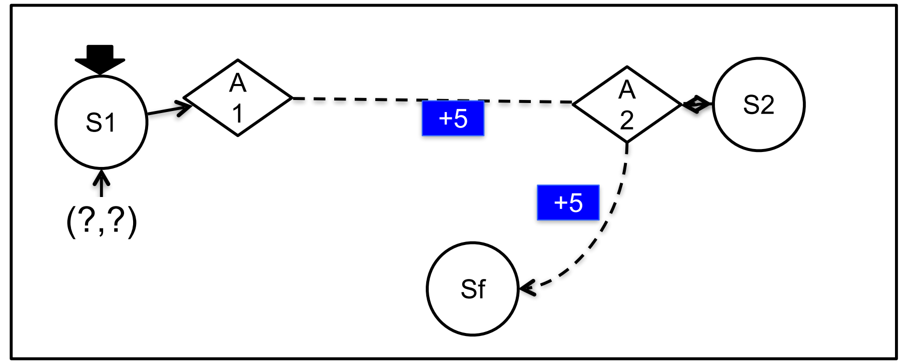
- **The interaction continues.**

Flight Booking with MDP (4/5)



- **The interaction continues.**
- **When the final state is reached, the task is completed and result is returned.**

Flight Booking with MDP (5/5)

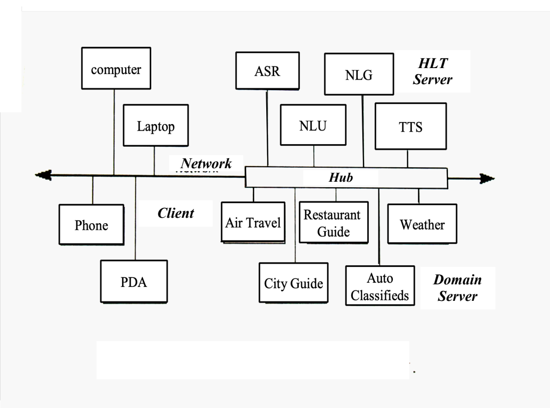


- **For the overall dialogue session, the goal is to maximize the total reward**

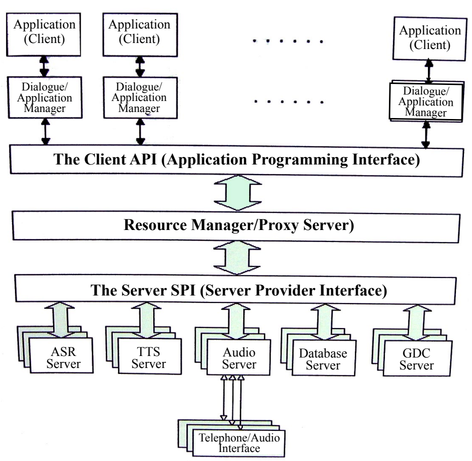
  $R = R1 + \ldots + Rn = 5 + 5$
- **Dialogue optimized by choosing a right action given each state (policy).**
- **Learned by Reinforcement Learning.**
- **Improved as Partially Observable MDP (POMDP)**

Client-Server Architecture

- **Galaxy, MIT**

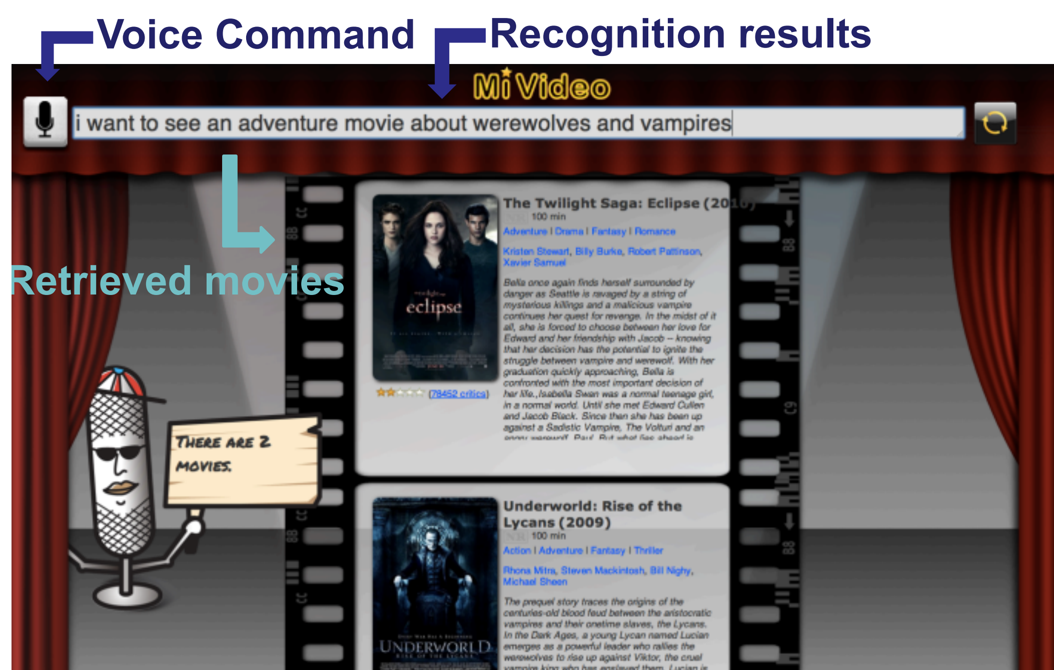


- **Integration Platform, AT& T**

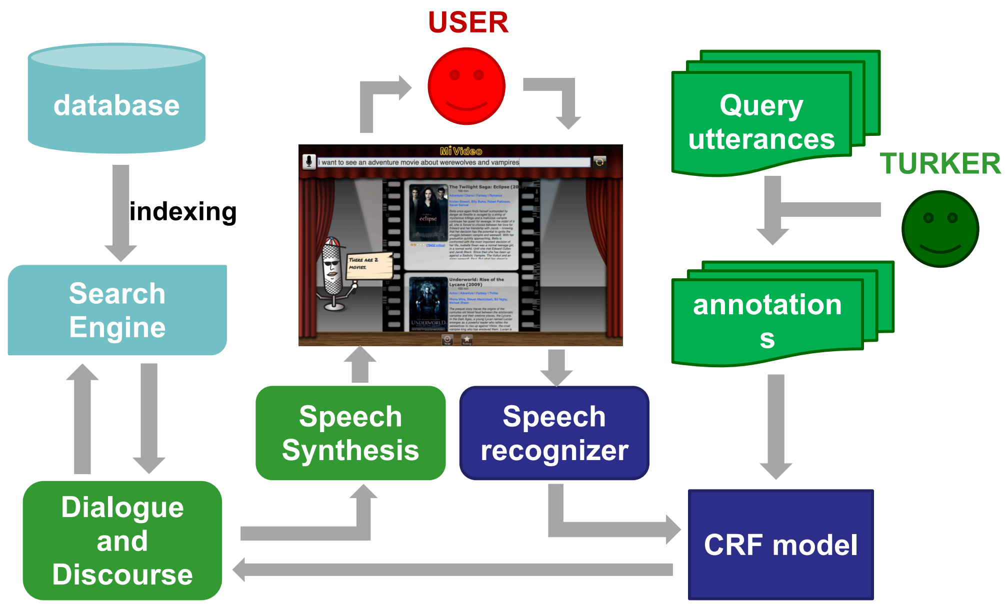


- **Domain Dependent/Independent Servers Shared by Different Applications/Clients**
  - reducing computation requirements at user (client) by allocating most load at server
  - higher portability to different tasks

An Example: Movie Browser



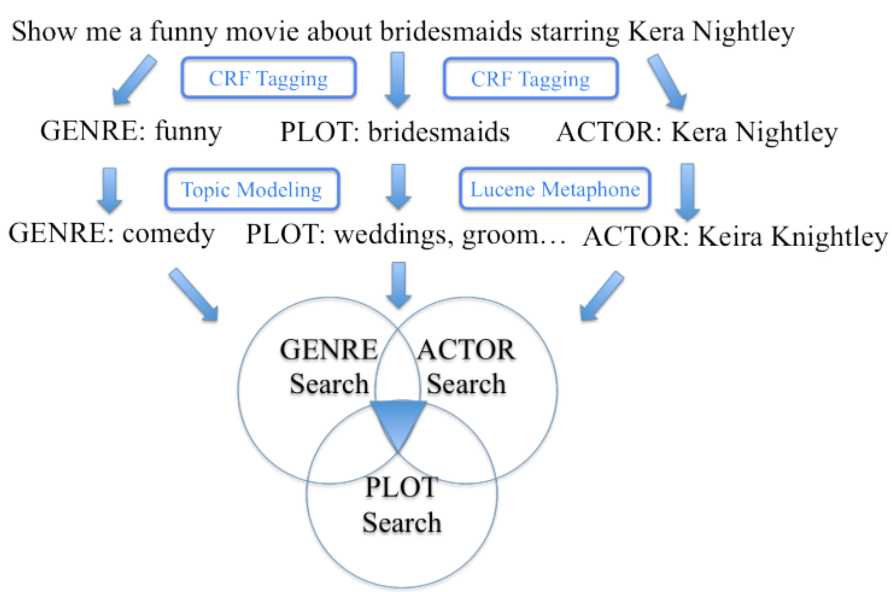
Flowchart



Semi-CRF for Slot Filling

- **Input data: user's query for searching movie**
- **Ex: Show me the scary movie**
- **Output: label the input sentence with "GENRE", "PLOT" and "ACTOR"**
- **Topic modeling**
  - Data sparsity → difficult to match terms exactly
  - Ex. "funny" and "comedy"
  - Use Latent Dirichlet Allocation (LDA) for topic modeling
- **Handling misspelling**
  - Convert query terms to standard phonemes
  - Search by pronunciations instead of spellings

Example



References for CRF

- **References:**

  - Jingjing Liu, Scott Cyphers, Panupong Pasupat, Ian Mcgraw, and Jim Glass, **A Conversational Movie Search System Based on Conditional Random Fields** , Interspeech, 2012
  - J. Lafferty, A. McCallum, and F. Pereira. **Conditional random fields: Probabilistic models for segmenting and labeling sequence data**, In Proc. of ICML, pp.282-289, 2001
  - Wallach, H.M., **Conditional random fields: An introduction**, Technical report MS-CIS-04-21, University of Pennsylvania 2004
  - Sutton, C., McCallum, A., **An Introduction to Conditional Random Fields for Relational Learning**, In Introduction to Statistical Relational Learning 2006

References for CRF

- **References:**
  - Sunita Sarawagi, William W. Cohen: **Semi-Markov Conditional Random Fields for Information Extraction**. NIPS 2004
  - Bishan Yang and Claire Cardie, **Extracting Opinion Expressions with semi-Markov Conditional Random Fields**, EMNLP-CoNLL 2012
- **Toolkits:**
  - CRF++ (http://crfpp.googlecode.com/svn/trunk/doc/index.html)
  - CRFsuite (http://www.chokkan.org/software/crfsuite/)