Dialogue Strategy

• Dialogue strategy is more art than engineering or science

- Should the system ask explicit confirmations after each response?
- Should the system use open-ended questions to begin with?
- What order should the system ask parameters in?

Overview

⇒ Intro
⇒ Dialogue Quality
⇒ Example
⇒ Formulating Strategy
⇒ MDP
Use human-human data?

- Collect a whole bunch of human-human dialogues in domain

But...

- Does the system expect the system to behave like another person?
- Are people always using optimal strategy?
- System will have different difficulties
  - Its memory will be perfect
  - Its understanding won't be as good

Many advances in SLT come from data driven approaches

Data Driven Approaches

- Speech recognition uses a corpus of examples from which to find good speech synthesis uses data to estimate word probabilities
- Speech recognition uses data to build acoustic models
- Sufficient approaches to parsing have also been proposed

So, let's gather some data!

Do the same for dialogue strategy!
Plan of Attack

- Have computer system interact with simulated user
- Try out a lot of different combinations of moves
- See what moves are good
- Use good moves to build optimal dialogue strategy

Use Wizard-of-Oz approach?

- Use Wizard-of-Oz approach?
  - Build interface of a system but have a person (wizard) drive it
  - All interface will be exactly like future system will be
  - All of the prompts are pre-recorded
  - Must collect enough dialogues to cover all possible situations that you could get into
  - Collect dialogues with users
  - Train a wizard to drive system
  - Collect dialogues with users
  - Build interface of a system but have a person (wizard) drive it

- If you change interface, need to collect a new corpus
  - Must collect enough dialogues to cover all possible situations that you could get into
  - If you change interface, need to collect a new corpus
  - Difficult to train a wizard to make same kind of mistakes system will make
  - Users don’t know they are interacting with a wizard

- But...
  - Difficult to train a wizard to make same kind of mistakes system will make
  - Must collect enough dialogues to cover all possible situations that you could get into
Quantifying how good a system is in carrying out a dialogue...

- How many database queries were made?
- Did it get right value for each piece of data?
- How well did the system get all of the pieces?
- Did the user enjoy the interaction?
- How long was the interaction?

We will measure the cost of a dialogue in terms of the following:

- The lower the cost, the better it is.
- Need to measure how good a system is in carrying out a dialogue.

Overview

• Intro
• Example
• Formulating Strategy
• MDP
Overview

• Intro
• Dialogue Quality
  ⇒ Example
• Formalizing Strategy
• MDP

Mathematical Formula

We will talk later about how to figure out the weights between the various dimensions.

Note that we have expressed the overall cost as a tradeoff:

\[ C = \sum W_i C_i \]

- Let \( \langle C \rangle \) be expected value of \( C_i \).
- Want a system that works good on average; rather than just in one case.
- But, want to average this over a bunch of dialogues.

Mathematical Formula

- Say we have \( n \) dimensions, each has a cost (quality) \( C_i \).
- Overall cost of a dialogue is \( C = \sum W_i C_i \).
- Each dimension will have a different impact on quality.
- Note that we are making a linearity assumption.
Metric

• Note: other formulations of dialogue quality are possible

\[ C = \frac{1}{N} \sum_{i=1}^{N} \left( \lambda_i + \sum_{e \in E_i} \lambda_e + \sum_{f \in F_i} \lambda_f \right) \]

- \( f \)_i number of filled slots
- \( e \)_i number of errors in the obtained values
- \( f \)_i number of interactions (turns)

\( \sum_{i=1}^{N} \left( \lambda_i + \sum_{e \in E_i} \lambda_e + \sum_{f \in F_i} \lambda_f \right) = C \)

Application

• Domain
  - Form-filling dialogue
  - Fill in the day and month

• Actions
  - Ask question asking for the value of the day
  - Ask question asking for the value of the month
  - Open-ended question asking for the value of the date
  - Ask question asking for the value of the day

• Actions (continued)
  - Fill-in-the-day and month
  - Form-fill dialogue

• Domain

(\( \tilde{d} < \beta \tilde{d} \) where \( \beta \tilde{d} \) is the length of the slot)

Assume probability of misrecognition in asking

- final action, closing the dialogue
- give question asking for the value of the date
- open-ended question asking for the value of the date
- give question asking for the value of the day
- give question asking for the value of the day
- give question asking for the value of the day
- give question asking for the value of the day
Which Strategy is Best?

• Is strategy 1 better than strategy 3?

\[ C_1 < C_3 \] (1)

\[ W_{i} + 2W_p < 3W_i + 2W_e \] (2)

\[ 2W_f < 2W_i + 2W_p \] (3)

\[ 2W_{id} + 2W_i > fW_m + W_i \] (4)

\[ 2f_{id} > fW_m \] (5)

Assuming \( W_e > 0 \)

- It is if the error rate is too high

Three Alternate Strategies
Dialogue Actions

- We could have treated each part as a separate action

  - System's recognition of what the user said
  - System's recognition of what the user said
  - System's interpretation of the user's response
  - System will only recognize the possible legal answers for the prompt
  - User responding

  Each action will include:

  - There are a set of actions

Dialogue Actions

- MDP
- Formalizing Strategy
- Example
- Dialogue Quality
- Intro
Alternate Set of States

- Have different states for how confident the system is in the user's response.
- Have different states for how many times the user has given a value for a particular prompt.
- Have different states for whether the user has said the same value for a particular prompt.

⇒ Current state should capture everything that is relevant for system to decide which action to perform for a particular prompt.

Dialogue States

- What different states could the system be in?

- For earlier example:
  Special final state (d=0, m=0)
  State where day is known, but month is not
  State where month is known, but day is not
  State where neither month nor day is known

- Set of states is up to the dialogue designer.

- Different states could system be in?
Dialogue Strategy

- Dialogue strategy: what action should be done in each state
- For day-month example, we might want to put restrictions on strategies
  - Always start from the initial state
  - The state that we designated as initial is final. No actions done in that state.
  - Ensure that last action is \( A_f \) whose effect is going to the final state.

- Strategy 3 from example:
  - In state \( d=0 \ m=0 \), perform \( A_d \)
  - In any other state with \( m=0 \), perform \( A_m \)
  - In any other state, perform \( A_f \)

States and Actions

- Doing an action in a certain state will take the system to another state
- Of course, not a deterministic move
- Don't know if we will correctly recognize response (see earlier example)
- Note that \( P(s_{t+1} | s_0, s_1, \ldots, s_{t-1}, a_0, a_{t-1}, \ldots, a_{t-1}) \) will be zero for some combinations
- If \( s_t \) indicates we know the day, so must \( s_{t+1} \)
- Model the result as \( P(s_{t+1} | s_t, a_t) \)

"If I have any other state where I want to put restrictions on strategies, what action should be done in each state?"
Overview

• Intro
• Dialogue Quality
• Example
• Formalizing Strategy

⇒ MDP

More Complex Dialogue Strategies

For our day-month example, and with the set of states that we have, there are really just 4 good strategies.

- Would need extra states, so as to indicate if there is good or poor confidence in recognition
- Others can be easily shown to be worse
- How do we model confirmations?

CSE530 Class RL: 22
© P. Heeman, 2003
Transition Probability Distribution

For our example:

- When in initial state $d=0$ and $m=0$ and ask which month $A_m$, we'll go to some state with $d=0$ and $m > 0$, but don't know which.
- Depends on how user responds and whether system correctly interprets it.

Describe next state probabilistically:

$$\Pr(s_{t+1} | s_t, \ldots, s_0, a_t, \ldots, a_0)$$

+ How system determines user's response depends on which state it was in and
+ Depends on how user responds and whether system correctly interprets it.
+ Will go to some state with $d=p$ and $m=q$ if don't know which.
+ When in initial state with $d=0$ and ask which month $A_m$.

For our example:

Assumption 1: $\Pr(s_{t+1} | s_t, \ldots, s_0, a_t, \ldots, a_0) = \Pr_T(s_{t+1} | s_t, a_t)$

Too many strategies

- Even for our month-date setup, there are lots of strategies.
- In 410 states, can do one of three actions $\Rightarrow$ 3$^{410}$ strategies.
- In 410 states, can do one of three actions, so need to do each strategy.
- Thus, user will behave differently each time, so need to do each strategy.
- Plus, user will behave differently each time, so need to do each strategy.
- Need to determine which actions $N$ states, will be $N^N$ strategies.
- If there are $N$ actions and $N^N$ states, will be $N^N$ strategies.

Instead:

- Piece together the parts into a scenario.
- Learn which parts of strategies work best.
- Run different strategies.

Can determine which pieces of Markov Decision Process (MDP).
Cost Probability Distribution

Cost of interactions $C_i$ can be distributed:

- $W_i$ for each action performed
- Other costs just associated with a complete dialogue, such as $C_f$ and $C_e$

For computing the cost of (at least) the end state

- $c_t$ is the change in cost of the dialogue from going from state $s_t$ to $s_{t+1}$ when $T_F$ is the time step when the final state $s_F$ is reached

Distribute cost of dialogue $C$ over course of dialogue

\[
\Pr(\{c_t\mid s_t, \ldots, s_0, a_t, \ldots, a_0\})
\]

Assumption 2:

\[\Pr(\{c_t\mid s_t, \ldots, s_0, a_t, \ldots, a_0\}) = \Pr(\{c_t\mid s_t, a_t\})\]

If it is usually straightforward to spot violations of this assumption

Our current state formulation already takes care of this

- After we know the month, we better continue to know the month
- After we know the day, we better continue to know the day

Problem for our day-month example:

History of previous states and actions cannot affect $s_{t+1}$

Enough States for Assumption 1?

\[\text{CSE560 Class RL: 25}\]

\[\text{CSE560 Class RL: 26}\]
Enough States for Assumption 2?

- History of previous states and actions cannot effect $c_t$

- Problem for our day-month example:
  - When we get to the state $d > 0$ and $m > 0$ and perform $A_f$ to get to the final state, $c_t$ depends on $\langle C_e \rangle$, which depends on whether we asked $A_dm$ or $A_d$ and $A_m$.
  - But this information is not captured in what state we are in.

- To satisfy assumption 2, might need to reformulate state space:
  - Add more states to capture whether day was answered from "what day" and month was answered from "what month".
  - Add more states to capture whether day was answered from "what day" and month was answered from "what month".

- To satisfy assumption 2, might need to reformulate state space:
  - Add more states to capture whether day was answered from "what day" and month was answered from "what month".

- To satisfy assumption 2, might need to reformulate state space:
  - Add more states to capture whether day was answered from "what day" and month was answered from "what month".

Usefulness of Incremental Costs

- For dialogue systems, almost always want to penalize shorter ones.
  - Waiting until the end of the dialogue allows us to get around this problem.
  - Distribution of these costs throughout the dialogue allows us to get around this problem.
  - Waiting until the end of the dialogue allows us to get around this problem.
  - Waiting until the end of the dialogue allows us to get around this problem.
  - Waiting until the end of the dialogue allows us to get around this problem.
  - Waiting until the end of the dialogue allows us to get around this problem.

- Waiting until the end of the dialogue allows us to get around this problem.
  - Waiting until the end of the dialogue allows us to get around this problem.
  - Waiting until the end of the dialogue allows us to get around this problem.
  - Waiting until the end of the dialogue allows us to get around this problem.
  - Waiting until the end of the dialogue allows us to get around this problem.
  - Waiting until the end of the dialogue allows us to get around this problem.
Another Example

- Should system remember which pieces of information are confirmed?
- System should learn to do these depending on ASR confidence
  - Have actions to confirm each piece of information
    - Should system remember the confidence for each filler?
  - Should include ASR confidence in user's current answer
  - Domain: set values for 5 slots

Example Costs

- $c(\ast, A_d) = c(\ast, A_m) = c(\ast, A_{dm}) = W_i$
- Intermediate states
  - $c(s, A_f) = 1 W_i + 2 W_f$ with probability 1
    - Going to the final state right from the initial state
  - For all states $s$ with only one variable assigned with a simple
    question (say $d > 0$, $m = 0$, $q_d = 1$, $q_m = 0$)
    - $c(s, A_f) = 1 W_i + W_f$ with probability $p^2$
      - $c(s, A_f) = 1 W_i + 1 W_f$ with probability $1 - p^2$

Example Costs
Why RL

For each state and action, determine the cost to get to the final state

- RL allows you to determine an optimal policy
- Good

Do not know until the end whether a sequence of actions was

cost of getting to the end

Need to determine action to do in each state that gives the lowest

- Different costs for doing an action in a certain state
- From any one state, only a subset of states are reachable
- Some states are reachable and some are not
- Distribution
- Connected by a probability
- Where you will go to after
- Cannot specify exactly
- States as a graph
- Visualize the dialogue

Visualizing an MDP