Dialogue Strategy

- But how do we know if we have a good order?
- In ISU, this is hand-coded by order of the rules.
- Should order of parameters in the system be decided?
- Should the system ask an open-ended question to begin with?
- Should the system use explicit confirmations after each user response?

Dialogue strategy is more art than engineering or science.

Overview

- Intro
- Formalizing Strategy
- Example
- MDP
- Dialog Quality
- Dialogue Strategy

Use human-human data?

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

Data Driven Approaches

- Many advances in SLS come from data driven approaches
- Speech recognition uses data to build acoustic models
- Speech recognition uses data to estimate word probabilities
- Speech synthesis uses a corpus of examples from which to find good pieces to concatenate
- Statistical approaches to parsing have also been proposed
- Do the same for dialogue strategies!

... +
+ His memory will be perfect
+ His understanding won't be as good
- System will have different difficulties
- Do users expect system to behave like another person?
- Are people always using optimal strategies?

But...

So, let's gather some data!
Plan of Attack

- Use good moves to build optimal dialogue strategies.
- Try out a lot of different combinations of moves.
- Certainly true for form-filling applications.
- Assumption that user's behavior is simpler than system's.
- Have computer system interact with simulated user.

Use Wizard-of-Oz approach?

- Build interface of a system, but have a person (wizard) drive it.
- Users don't know they are interacting with a wizard.
- Collect dialogues with users.
- Train a wizard to drive system.
- All of the prompts are pre-recorded.
- Interface will be exactly like the human system will be.
- If you change interface, need to collect a new corpus.
- Difficult to train a wizard to make same kind of mistakes system will.
- Must collect enough dialogues to cover all possible situations that you.

But...

- Difficult to train a wizard to make same kind of mistakes system will.
- Must collect enough dialogues to cover all possible situations that you.
- Users don't know they are interacting with a wizard.
- Collect dialogues with users.
- Train a wizard to drive system.
- All of the prompts are pre-recorded.
- Interface will be exactly like the human system will be.
- If you change interface, need to collect a new corpus.
- Difficult to train a wizard to make same kind of mistakes system will.
- Must collect enough dialogues to cover all possible situations that you.

But...

- Difficult to train a wizard to make same kind of mistakes system will.
- Must collect enough dialogues to cover all possible situations that you.
- Users don't know they are interacting with a wizard.
- Collect dialogues with users.
- Train a wizard to drive system.
- All of the prompts are pre-recorded.
- Interface will be exactly like the human system will be.
- If you change interface, need to collect a new corpus.
- Difficult to train a wizard to make same kind of mistakes system will.
- Must collect enough dialogues to cover all possible situations that you.
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 did the interaction take?
- Might quantify it in terms of the following...
  - How long did the interaction take? The lower the cost, the better it is.
  - How well did the system get all of the pieces?
  - Did it get right value for each piece of data?
  - How many database queries were made?

Need to measure how good a system is in carrying out a dialogue.
Overview

Intro

Dialogue Quality

⇒ Example

Formalizing Strategy

MDP

Mathematical Formula

Say we have \( n \) dimensions, each has a cost (quality) \( C_i \).

- Each dimension will have a different impact on quality.
- Want a system that works good on average, rather than just in one case.
- Let \( \langle C_i \rangle \) be expected value of \( C_i \).
- Want to average this over a bunch of dialogues.
- Overall cost of a dialogue is \( C = \sum W_i C_i \).

Note that we have expressed the overall cost as a tradeoff between the various dimensions.

- Overall cost of system is \( C = \sum W_i \langle C_i \rangle \).
- Let \( \langle C \rangle \) be expected value of \( C \).

But, want to average this over a bunch of dialogues.

- 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.

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

- Note that we are making a linearity assumption.
Note: other formulations of dialogue quality are possible

- \( \langle N_i \rangle \) number of turns
- \( \langle N_e \rangle \) number of errors in the obtained values
- \( \langle N_f \rangle \) number of unfilled slots

\[
\langle N \rangle = \langle N_i \rangle + \langle N_e \rangle + \langle N_f \rangle
\]

**Metric**

**Application**

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

**Actions**
- \( A_m \) question asking for the value of the month
- \( A_d \) question asking for the value of the day
- \( A_{dm} \) open-ended question asking for the value of the date
- \( A_f \) final action, closing the dialogue

Assume probability of misrecognition in asking
- \( p_1 \) probability of getting the month wrong
- \( p_2 \) probability of getting the day wrong

\( p_1 > p_2 \) since the month and day are in the same sentence increasing ASR difficulty
How were costs determined?

• Strategy 1:
  - Just do action \( A_f \) \( N_i = 1 \) (1 interaction) & \( N_f = 2 \) (2 unfilled slots)

• Strategy 2:
  - First ask date (day & month) then end-action
  - \( N_i = 2 \) (2 interactions)
  - \( N_f = 0 \) (both slots filled)
  - \( N_e = 2 \), where \( p_1 \) is prob of ASR making an error in day or month slot
  - \( 1 \ast p_1 \) \( \text{prob of making an error in day} \)
  - \( 1 \ast (1 - p_1) \) \( \text{prob of just making an error} \)
  - \( 2 \ast p_1 \) \( \text{prob of making an error in both} \)
  - \( C = 3 \cdot p_1 + 2 \cdot (1 - p_1) \)

### Three Alternate Strategies

- Strategy 2:
  - \( C = 3 \cdot p_1 + 2 \cdot (1 - p_1) \)
  - \( C = 5 \cdot p_1 \)
  - \( C = 1 \cdot p_1 \)

- Strategy 1:
  - \( C = 3 \cdot p_1 + 2 \cdot (1 - p_1) \)
  - \( C = 5 \cdot p_1 \)
  - \( C = 1 \cdot p_1 \)

- Strategy 1:
  - \( C = 3 \cdot p_1 + 2 \cdot (1 - p_1) \)
  - \( C = 5 \cdot p_1 \)
  - \( C = 1 \cdot p_1 \)
Overview

• Intro
• Dialogue Quality
⇒ Formalizing Strategy
• Example

Which Strategy is Best?

- Is strategy 1 better than strategy 3?
- Is the expected cost of strategy 1 less than that for strategy 3?

\[
\begin{align*}
W_f + 2W_e &< 3W_i + 2p^2W_e \\
2W_f &< 2W_i + 2p^2W_e \\
W_f - W_i &< p^2W_e
\end{align*}
\]

Assuming \(W_e > 0\)

Note: in practice cannot derive the exact cost of a strategy
**Dialogue States**

- What different states could system be in?
- For earlier example:
  - State where neither month nor day is known (d=0, m=0)
  - State where month is known, but day is not
  - State where day is known, but month is not
  - State where month and day are known (d=0, m=0)
  - Special final state (d=-1, m=-1)

- Set of states is up to the dialogue designer

**Dialogue Actions**

- There are a set of actions
- Each action will include:
  - System's prompt
  - User responding
  - System's recognition of what the user said
- System will only recognize the possible legal answers for the prompt
- System's interpretation of the user's response
- System's recognition of what the user said
- System's interpretation of the user's interpretation of the user's response

- However, coarse granularity makes it hard to model barge-in
- Silly to learn "we interpret it" as a separate action but
- Coarse granularity simplifies things

- We could interpret each part as a separate action, but...
States and Actions (continued)

• Example: If user asked same question twice
  + if $a_5$ is ask month, user answers February, $m = 2$ in $s_6$
  + if $a_7$ is also ask month, user answers March, $m = 3$ in $s_8$

- Current state only captures value of month and not how many times asked and whether the value changed
- History of states and actions captures how many times asked and whether times asked

Note that $\Pr(s_{t+1}|s_t, s_{t-1}, ..., s_0, a_t, a_{t-1}, ...) = 0$ for some combinations

Example: If user asked same question twice

States and Actions
Dialogue Strategy (continued)

- In any other state, perform \( A_f \)
- In any other state with \( n=0 \), perform \( A^n \)
- In state \( d=0 \), perform \( A_f \)

Strategy 3 from example:

- Only consider states reachable from initial state and in that state
- Don't consider any actions out of the final state, since we automatically ensure \( A_f \) will be done
- For month-day example, any reasonable strategy:

reasonable strategies are reasonable

This example is simple enough that we can analyze what

How many are there for this example?

For every dialogue state, specifies what action system will perform

Dialogue strategy can be viewed as a mapping:
Alternate Set of States

- Current state should capture everything that is relevant for system to decide which action to perform.
- Have different states for how confident system is in user response.
- Have different states for how many times user has given a value for a particular prompt.
- Have different states for whether user has said the same value for a particular prompt.
- As system can just use current state to pick its action.

More Dialogue Strategies

- For our month-day example, and with the set of states that we have, there are many possible strategies.
- But only 4 good strategies.
- Others can be easily shown to be worse.
- Since just a few reasonable strategies, could simulate each strategy a large number of times to determine its average cost.
- Need to simulate a large number of times to cover how any possible user might behave, and how their actions might be misinterpreted.
Too Many Strategies

• Even for our month-day task, there are lots of strategies—
  Of course we know there are really just 4 good ones, but a data-driven
  approach would need to determine this itself.

⇒ Brute-force approach won't work!

Expected (average) cost
be misunderstood, so need to do each strategy many times to figure out
Plus, each user will behave differently (different slot values), and mutually
If A\_actions and A\_states, will be A\_states, so 40
Hence, non-trivial states can do one of three actions
410 strategies

[Overview]

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

© 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:
  \[
  \Pr(s_{t+1} | s_t, ..., s_0, a_t, ..., a_0)
  \]
  - Depends on how user responds and whether system correctly interprets it.
  - 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.

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

Alternative

- Instead, run different strategies.
- Learn which of strategies work best.
- Piece together the parts into a strategy.
- Can determine which pieces fit Markov Decision Process (MDP).
- Instead, run different strategies.
Cost Probability Distribution

\[ C = \langle \sum_{t=0}^{T_F} c_t \rangle \]

where \( T_F \) is the time step when the final state is reached.

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 \( C_e \) plus the change in cost of the dialogue from state \( s_t \) to \( s_{t+1} \) where \( t \) is the time step when the final state is reached.

Distribute cost of dialogue \( C \) over course of dialogue.

If it is usually straightforward to spot violators of this assumption:

\[ \text{enough states for Assumption I?} \]

Enough States for Assumption I?

Problem for our month-day example?

History of previous states and actions cannot affect \( s_{t+1} \).
Enough States for Assumption 2?

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

  - Problem for our month-day example:
    - When we get to the state $d > 0 \land m > 0$ and perform $A_f$ to get to the final state, $c_t$ depends on $\langle C_e \rangle$, which depends on whether we asked $A_d m$ 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'.
      - Include in state $q_d = 1$ if day value is from asking 'what day', and $q_m = 1$ if month value is from asking 'what month'.

Usefulness of Incremental Costs

- For dialogue systems, almost always want to penalize longer dialogues.

  - So, final dialogue cost will include some weight times the number of interactions.
  - Waiting until the end of the dialogue:
    - States need to capture the number of interactions in order to meet assumption 2.
    - Willing until the end of the dialogue:
      - Distributing these costs throughout the dialogue allows us to get around this problem.
      - Will need much larger state space.

For dialogue systems, almost always want to penalize longer dialogues over shorter ones.

Usefulness of Incremental Costs
Another Example

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

Example Costs

- $c(\ast, A_d) = c(\ast, A_m) = c(\ast, A_{dm}) = W_i$
- Intermediate states
  - $c(s_I, A_f) = 1 W_i + 2 W_f$ 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 $d > 0, m = 0, q_d = 1, q_m = 0$)
    - $c(s, A_f) = 1 W_i + W_f + W_e$ with probability $p$
    - $c(s, A_f) = 1 W_i + 1 W_f$ with probability $1 - p$

Example Costs
Why RL

- Do not know until the end whether a sequence of actions was good
- RL allows you to determine an optimal policy

Cost of getting to the end
- Need to determine action to do in each state that gives the lowest cost of getting to the end
- 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

Visualization of MDP
- Cannot specify exactly what state you will go to after doing an action
- Controlled by a probability distribution
- Can only specify exactly what states are reachable