Lightly Supervised Learning of Procedural Dialog Systems

Svitlana Volkova
Pallavi Choudhury, Chris Quirk, Bill Dolan, Luke Zettlemoyer
Motivation

- Procedural dialog systems aim to assist users with a range of goals:
  - guide visitors through a museum (Traum et al., 2012)
  - teach students physics (Dzikovska et al., 2011)
  - enable interaction with a health care system (Morbini et al., 2012)

- Limitations:
  - require expensive manual engineering
  - substantial domain-specific task knowledge

Building scalable dialogue systems in an automated way with light supervision
Intent-Oriented Dialog

U: “I want to add page numbers and a title”
S: “Top or Bottom of the page?”
U: “Top”
S: “Please select page design from the templates”
(*System shows drop down menu*)
U: *User selects from menu*
S: “Enter header or footer content”
U: “C.V.”
S: “Task completed.”
Our Approach

I. Learn procedural dialog systems from task-oriented textual resources
   ✓ I. Query logs
   ✓ II. Instructional text

II. Use light non-expert supervision

III. No help of knowledge engineers or domain experts!
### Data: Query to Help Page Mapping

#### Queries

- help with page numbers in the header
- how to number some pages
- word adding page numbers
- how do you add numbering word format
- footer sections
- break page
- how do I page break
- manual page break
- inserting a manual page break
- add background
- add line background
- how to add a background
- delete background
- how do you delete the color background
- inserting comments
- insert a comment
- how to insert a comment in word
- how to add comments
- insert comment in word 2007
- how to delete comments in word
- delete comments function

#### Help Pages

**Add and format page numbers**

If your document has multiple pages, you can display the page number on each page. In most documents, page numbers appear in either the header or the footer of each page. Microsoft Office Word 2007 has many preformatted page number designs so that you can quickly insert headers or footers that display the page number.

**Add a page break**

When you're typing a document, Word automatically adds a page break at the end of each page, depending on where you set your margins. (You can also use section breaks to control the formatting of your document.)

However, if you want to add a page break somewhere else—for example, in the middle of the page so that later you can add a drawing or graphic—you can insert a manual page break. (You can always delete a page break that you've added but you can't remove one that Word automatically adds at the end of each page.)

1. Click where you want to start a new page.
2. Go to the Insert tab, then click Break.

**Insert or delete a comment**

You can insert a comment inside balloons that appear in the document margins. You can also hide comments from view.

A background or a page color is primarily used to create a more interesting background for online viewing. Backgrounds are visible in Web Layout view and Full Screen Reading view.

You can use gradients, patterns, pictures, solid colors, or textures for backgrounds. Gradients, patterns, pictures, and textures are tiled or repeated to fill the page. If you save a document as a Web page, the pictures, textures, and gradients are saved as JPG files and the patterns are saved as GIF files.

**Add a background to an online document**

- Change a document background
- Remove a background

---

**Insert a comment in word 2007**

1. On the Page Layout tab, in the Page Background group, click Page Color.
2. Do any of the following:
   - Click the color that you want under Theme Colors or Standard Colors.
   - Click More Colors and then click a color.
   - Click Fill Effects to add special effects, such as gradients, patterns, or textures.
Other sources of instructional text:
- recipes that describe how to cook meals
- software help web pages that describe how to achieve goals
  www.ehow.com and www.wikianswers.com
Data II. Query Logs

Search query logs = User intent statements

- Queries mapped to Microsoft Office Help Pages
  - High frequency pain points (>1K queries per page)

- Other sources of user intents:
  - Crowdsourcing or popular applications

- This is not dialog data! No sequences of utterances!
- Data omits slot fillers e.g., for the goal "add a border"
  - color, style typically left unspecified
Key Steps

I. Learning to convert the textual knowledge into procedural knowledge
   - Building Dialog Trees from Instructions

II. Learning a dialog manager using classification-based approach
   - Understanding Initial Dialog Turns
   - Understanding Subsequent Dialog Turns
Building Dialog Trees from Instructions

- **Input:**
  - help pages and queries

- **Output:**

  **Step 1. Extraction**
  - for every page extract a topic, goals and instructions

  **Step 2. User Action Classification**
  - classify user actions and identify system actions
  - construct dialog trees
Input: Instructions and Queries

 Queries

- help with page numbers in the header
- how to number some pages
- microsoft word adding page numbers
- how do you add numbering in ms word
- word format footer sections
- how do you add a page number on the right and header on the left
- insert page numbers not available
- how to insert page numbers on both side of the pages

 Instructions

[Diagram showing steps to create a header or footer that contains only the page number]

[Diagram showing steps to place the page number in the side margin of the page]
Output: A Dialog Tree

Add header or footer content to a preformatted page number design

Add and format page numbers

Include Page or Page X of Y with the page number

1.1
Add header or footer content to a preformatted page number design

1.2
Add and format page numbers

1.3
Include Page or Page X of Y with the page number

1.4
Binary

a_u: Binary

a_s: “Top of Page or Bottom of Page?”

q_g

a_u: Selection

a_s: Popup

q_a

a_u: Input

Execute
Step 1. Extraction

**Office Help Page**

**ADD AND FORMAT PAGE NUMBERS**

**CREATE A HEADER OR FOOTER THAT CONTAINS ONLY THE PAGE NUMBER**

1. On the Insert tab, in the Header & Footer group, click Page Number.
2. Click Top of Page or Bottom of Page, depending on where you want page numbers to appear in your document.
3. Choose a page number design from the gallery of designs.

**INCLUDE PAGE OR PAGE X OF Y WITH THE PAGE NUMBER**

1. On the Insert tab, in the Header & Footer group, click Page Number.
2. Click Top of Page, Bottom of Page, Page Margins, or Current Position depending on where you want page numbers to appear.
Step 2. User Action Classification

- Data (manually labeled):
  - 30 topics with >1K queries, 76 goals and 246 instructions, ~ 3 instructions per goal

\[
p(a_u | i, \theta) = \frac{e^{\theta \cdot \phi(a_u, i)}}{\sum_{a_u'} e^{\theta \cdot \phi(a_u', i)}}
\]

Classify user actions **automatically** with **limited** annotation effort

<table>
<thead>
<tr>
<th>Features</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline 1: Majority</td>
<td>0.53</td>
</tr>
<tr>
<td>Baseline 2: Heuristic</td>
<td>0.64</td>
</tr>
<tr>
<td>Ngrams</td>
<td>0.89</td>
</tr>
</tbody>
</table>
Key Steps

I. Learning to convert the textual knowledge into procedural knowledge
   ✓ Building Dialog Trees from Instructions

II. Learning a dialog manager using classification-based approach
   ❑ Understanding Initial Dialog Turns
   ❑ Understanding Subsequent Dialog Turns
Understanding Initial Dialog Turns

- **Input:**
  - dialog trees with topics, goals, actions and queries
  - new query $q$

- **Output:**
  - an initial dialog state $s_0$ that maximizes $p(s_0 | q, \theta)$
  - predict the deepest relevant node in the dialog tree
Query To Dialog State Mapping

**t_1 queries:**
- page number
- numbering pages
- how do you number pages
- how do you do page numbering

**g_{1.1} queries:**
- number page with text
- page numbering with text prefix
- how do you put your name and a page number on each page

**a_{1.1} queries:**
- headers page number
- inserting page number in header
- page numbers on bottom of page
- page numbering at bottom page

**t_1**

\[
p(s_0|q, \theta) = \frac{e^{\sum_i \theta_i \phi_i(s_0, q)}}{\sum_{s_0'} e^{\sum_i \theta_i \phi_i(s_0', q)}}
\]

q_1: how to number pages
q_2: add content with page number
q_3: set a page number in a footer
Results: Understanding Initial Dialog Turns

- **Lexical features**: Query n-grams + stems + no stop words
- **State features**:
  - TFIDF 10 best scores and query q overlap with:
    - **QueryOverlap**: a set of queries from a node
    - **QueryHistoryOverlap**: queries from the parent nodes
- **Data**: ~1K query to node mappings (80/20 splits)
- 5 - 10 queries per every topic, goal and less for action nodes

<table>
<thead>
<tr>
<th>Feature Types</th>
<th>Topic</th>
<th>Goal</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline 1: Random</td>
<td>0.10</td>
<td>0.04</td>
<td>0.04</td>
</tr>
<tr>
<td>Baseline 2: TFIDF 1 Best</td>
<td>0.81</td>
<td>0.21</td>
<td>0.45</td>
</tr>
<tr>
<td>Lexical (L) n-grams + stop words + stems</td>
<td>0.92</td>
<td>0.66</td>
<td>0.63</td>
</tr>
<tr>
<td>L + TFIDF 10Best + QueryOverlap</td>
<td>0.95</td>
<td>0.72</td>
<td>0.69</td>
</tr>
<tr>
<td>All above + QueryHistoryOverlap</td>
<td>0.96</td>
<td>0.73</td>
<td>0.71</td>
</tr>
</tbody>
</table>
Understanding Subsequent Dialog Turns

- **Input:**
  - a query $q$, system action from previous dialog state and a new query $q'$

- **Output:**
  - a new dialog state $s'$ at any level in the dialog tree:

$$p(s' | q', s, \theta) = p(s' | q', q, a_s, \theta).$$
Dialog State Update Classification

Start

ADD AND FORMAT PAGE NUMBERS

S: “Top or Bottom of the page?”

U: “Top”

t_{1.1}

S: “Please select page design from the templates”

U: Select “CV”

Execute

APPEND

INSERT A SECTION BREAK

S: “Enter header or footer content”

U: “C.V.”

S: “Task completed.”
Dialog State Update Classification

U: I want to add page numbers and a title

U: “I want to add page numbers and a title”
U: “add page X of Y” ……

OVERRIDE

ADD AND FORMAT PAGE NUMBERS

INSERT A SECTION BREAK

Execute
Dialog State Update Classification

\[
p(s' | q', q, a_s \theta) = \frac{e^{\sum_i \theta_i \phi_i(s', q', a_s, q)}}{\sum s'' e^{\sum_i \theta_i \phi_i(s'', q', a_s, q)}},
\]

**Problem Definition**

We are given the previous dialog state \(s\) and the previous user utterance \(q\). We learn a linear classifier that models \(\theta\) which pairs a node from the dialog tree with a query \(q\) we are interpreting. This disambiguation allows us to find a new dialog state \(s'\) given the previous state and new query \(q\). Ideally, dialog systems should update their state when transitioning from a node to its new state parent \(s' = \theta(q, s)\). Dialogue state updates for topic, goal, and action nodes. We also developed a classifier model for mapping followup queries to dialog tree nodes while maintaining the current goal and rarely resetting.

**Definition 1.**

A transition from an action node must be to a different action node in the same dialog tree etc. An override action defines a dialog system action which pairs a node from the dialog tree with a query \(q\) that triggers an override when the user's intent is to change a goal within the current interaction. An append action defines a dialog system action which pairs a node from the dialog tree with a query \(q\) that triggers an append when the user's intent is to add page numbers in a different topic subtree, etc. A reset action defines a dialog system action which pairs a node from the dialog tree with a query \(q\) that triggers a reset when the user's intent is to restart the dialog system, e.g. "I want to add page numbers and a title." A dialog state update model to maximize a dialog state distribution over all nodes in the dialog network: \(p(s' | q', q, a_s \theta)\) is appropriate when the user's intent is to change a goal within the current interaction.

**Definition 2.**

We simulate previous state and new query \(q\) and get a corresponding system action \(a\). We present more examples for add and format page numbers, and a query \(q\). We are given a network of dialog trees \(T\), a state update when transitioning from a node to its new state parent \(s' = \theta(q, s)\). We use a log-linear model to maximize a dialog state distribution over all nodes in the dialog network: \(p(s' | q', q, a_s \theta)\).

**Definition 3.**

An override action defines a dialog system action which pairs a node from the dialog tree with a query \(q\) that triggers an override when the user's intent is to change a goal within the current interaction. An append action defines a dialog system action which pairs a node from the dialog tree with a query \(q\) that triggers an append when the user's intent is to add page numbers in a different topic subtree, etc. A reset action defines a dialog system action which pairs a node from the dialog tree with a query \(q\) that triggers a reset when the user's intent is to restart the dialog system, e.g. "I want to add page numbers and a title." A dialog state update model to maximize a dialog state distribution over all nodes in the dialog network: \(p(s' | q', q, a_s \theta)\) is appropriate when the user's intent is to change a goal within the current interaction.

**Definition 4.**

We simulate previous state and new query \(q\) and get a corresponding system action \(a\). We present more examples for add and format page numbers, and a query \(q\). We are given a network of dialog trees \(T\), a state update when transitioning from a node to its new state parent \(s' = \theta(q, s)\). We use a log-linear model to maximize a dialog state distribution over all nodes in the dialog network: \(p(s' | q', q, a_s \theta)\).

**Experimental Setup**

We simulate previous state and new query \(q\) and get a corresponding system action \(a\). We present more examples for add and format page numbers, and a query \(q\). We are given a network of dialog trees \(T\), a state update when transitioning from a node to its new state parent \(s' = \theta(q, s)\). We use a log-linear model to maximize a dialog state distribution over all nodes in the dialog network: \(p(s' | q', q, a_s \theta)\).
Dialog State Update Classification

\[ p(s'|q', q, a_s \theta) = \frac{e^{\sum_i \theta_i \phi_i(s', q', a_s, q)}}{\sum_{s''} e^{\sum_i \theta_i \phi_i(s'', q', a_s, q)}} \]

**Problem Definition**

We are given a network of dialog trees containing a dialog state that summarizes the history of the current interaction. We also developed a classifier model for mapping state update actions in Table 1.1.2.

**Definition 3.**

A transition from an action node must be to a different node (from a topic to a goal node), another in its parent sibling node in the same distribution, etc. It could also be from an action node to another action node in a different goal in the same topic tree, etc.

**Definition 4.**

A reset action defines a dialog state update when transitioning from a node to its state. The previous user utterance contains the previous user utterance such as:

\[ p(s'|q', q, a_s \theta) = \frac{e^{\sum_i \theta_i \phi_i(s', q', a_s, q)}}{\sum_{s''} e^{\sum_i \theta_i \phi_i(s'', q', a_s, q)}} \]

\[ \phi(s, q, a, \theta) = \frac{e^{\sum_i \theta_i \phi_i(s, q, a, \theta)}}{\sum_{s'} e^{\sum_i \theta_i \phi_i(s', q, a, \theta)}} \]

\[ p(s'|q', q, a_s \theta) = \frac{e^{\sum_i \theta_i \phi_i(s', q', a_s, q)}}{\sum_{s''} e^{\sum_i \theta_i \phi_i(s'', q', a_s, q)}} \]

\[ \phi(s, q, a, \theta) = \frac{e^{\sum_i \theta_i \phi_i(s, q, a, \theta)}}{\sum_{s'} e^{\sum_i \theta_i \phi_i(s', q, a, \theta)}} \]

**Experimental Setup**

For each of the above, we evaluate relative to large volumes of real user interaction data. Our query log data is used when the user's intent is to restart the dialog and rarely resets. We simulate previous state and new query overlap with n-gram overlap with all queries from the old and new state parents, respectively. We sample a new query from another node which pairs a node from the dialog tree with a query we are interpreting. This disambiguates the previous state and new query overlap with a system prompt.

**Dialog State Update Classification**

We present more examples for the templates:

\[ p(s'|q', q, a_s \theta) = \frac{e^{\sum_i \theta_i \phi_i(s', q', a_s, q)}}{\sum_{s''} e^{\sum_i \theta_i \phi_i(s'', q', a_s, q)}} \]

\[ \phi(s, q, a, \theta) = \frac{e^{\sum_i \theta_i \phi_i(s, q, a, \theta)}}{\sum_{s'} e^{\sum_i \theta_i \phi_i(s', q, a, \theta)}} \]

**Report Performance for Two Types of Dialog Updates**

We learn a linear classifier that models \( \phi \) for each transition type. We also developed a classifier model for mapping state update actions in Table 1.1.2.
Dialog State Turn Simulation

- Data: 1.5K Simulated Dialog Turns
- Previous state $s$ to new query $q'$ pairs

- For every node/state $n$:
  - sample $q \sim (q_1^n \ldots q_m^n)$
  - get system action $a_s$
  - sample a new node $n'$ using prior
  - sample $q' \sim (q_1^{n'} \ldots q_m^{n'})$

![Graph showing Update Prior with categories Reset, Override, and Append with respective probabilities 0.7, 0.25, and 0.05]
Results: Understanding Subsequent Dialog Turns

- **Lexical features**: query n-grams + stems + no stop words
- **State features**: query $q'$ overlap with
  - $Q\text{Overlap} + P\text{Overlap}$: prev. query and system prompt
  - $S\text{QOverlap}$: queries from prev. state $s$
  - $S'Q\text{Overlap}$: queries from new. state $s'$

<table>
<thead>
<tr>
<th>Feature Types</th>
<th>Topic</th>
<th>Goal</th>
<th>Action</th>
<th>Append</th>
<th>Override</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexical (L)</td>
<td>0.92</td>
<td>0.76</td>
<td>0.78</td>
<td>0.90</td>
<td>0.89</td>
</tr>
<tr>
<td>L + QOverlap + POverlap</td>
<td>0.94</td>
<td>0.80</td>
<td>0.80</td>
<td>0.93</td>
<td>0.85</td>
</tr>
<tr>
<td>L + S’QOverlap</td>
<td>0.93</td>
<td>0.80</td>
<td>0.80</td>
<td>0.91</td>
<td><strong>0.90</strong></td>
</tr>
<tr>
<td>L + QOverlap + SQOverlap</td>
<td>0.94</td>
<td>0.81</td>
<td>0.81</td>
<td>0.91</td>
<td>0.88</td>
</tr>
<tr>
<td><strong>L + SQOverlap + S’QOverlap</strong></td>
<td><strong>0.95</strong></td>
<td><strong>0.84</strong></td>
<td><strong>0.83</strong></td>
<td><strong>0.94</strong></td>
<td>0.88</td>
</tr>
</tbody>
</table>
Key Steps

I. Learning to convert the textual knowledge into procedural knowledge
   - Building Dialog Trees from Instructions

II. Learning a dialog manager using classification-based approach
   - Understanding Initial Dialog Turns
   - Understanding Subsequent Dialog Turns

Does it really work?
Small Scale User Study

- Select 8 goals, 48 interactions total
- Ask 6 users to rate the system on a scale from 1 to 5

<table>
<thead>
<tr>
<th>User Satisfaction</th>
<th>Av. Score ± Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall satisfaction</td>
<td>3.42 ± 0.49</td>
</tr>
<tr>
<td>Questions were relevant</td>
<td>4.17 ± 0.41</td>
</tr>
<tr>
<td>System managed ambiguity</td>
<td>4.00 ± 0.89</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Objective Measures</th>
<th>Rates</th>
</tr>
</thead>
<tbody>
<tr>
<td>User achieved task</td>
<td>83%</td>
</tr>
<tr>
<td>System asked irrelevant questions</td>
<td>19%</td>
</tr>
<tr>
<td>System asked redundant questions</td>
<td>18%</td>
</tr>
</tbody>
</table>
User Study Error Analysis

- Missed key information in the input and asked redundant questions:
  
  U: “I want the background color to be yellow”
  S: “Choose a color for the document”
  U: “Yellow”
  S: “Task completed”

- Failed to correctly identify the intent and asked an irrelevant question:
  
  U: “I want to change the font face and size”
  S: “Choose the default font”
  U: “No, reduce the size of the text”
  S: *System Restarts*
Conclusions

- Construct dialog systems with light supervision automatically given only textual resources:
  - instructional text
  - search query click logs

Future Work:

- Scale to new domains!
- Increase the complexity of the dialog manager
- Explore the range of possible textual knowledge sources that could be incorporated
- Enable end users to author new goals by writing procedural instructions in NL
Thank you!

Contacts:
Svitlana Volkova – svitlana@jhu.edu
Pallavi Choudhury – pallavic@microsoft.com
Chris Quirk – chrisq@microsoft.com
Bill Dolan – billdol@microsoft.com
Luke Zettlemoyer – lsz@cs.washington.edu
Office Help Pages (Topics) and Queries – \( t_1 \ldots t_n \) and \( q_1 \ldots q_m \)
Goals and Instructions – \( q_1 \ldots q_k \) and \( i_1 \ldots I_l \)
Dialog trees – \( f_1 \ldots f_k \)

Query Paraphrases
- help with page numbers in the header
- how to number some pages
- adding page numbers
- how do you add numbering

- how to set a font size to stick in word
- default font
- how to set the font
- how do you change default font and font size in office?

Queries
- \( q_1 \)
- \( \ldots \)
- \( q_m \)

Topics
- \( t_1 \)
- \( \ldots \)

Goals
- \( g_1 \)
- \( \ldots \)
- \( g_k \)

Instructions
- \( i_1 \)
- \( \ldots \)

Dialog Trees
- \( f_1 \)
- \( \ldots \)
- \( f_n \)
Dialog State Update: From Topic Node

q: how do you number pages
q': set a page number in a footer
q: insert a comment
q': insert numbers in margins
q': add a background
q': add a comment “redo”

Execute

RESET APPEND

Start

t_1

g_1.1 g_1.2 g_1.4

Execute

a_1.2.1

Execute

a_21.3.1 a_21.3.2 a_21.3.3

g_21.1 g_21.2 g_21.3

t_21

......
Dialog State Update: From Goal Node

- **g1.1**: q': inserting page x of y
- **g1.3**: q': setting page x of y in footer
- **g1.4**: q': add a border
- **g6.1**: q': setting a default font
- **a6.4.1**: q': set default font and style

**Actions:**
- **Execute**
- **RESET**
- **OVERRIDE**
- **APPEND**

**Nodes:**
- Start
- t1
- t3
- t6
- t...

**Connections:**
- g1.1 → t1
- t1 → g1.3
- g1.3 → t3
- t3 → g1.4
- g1.4 → g6.1
- g6.1 → a6.4.1
- a6.4.1 → Execute
Dialog State Update: From Action Node

- $g_{1.1}$: Insert page numbers
- $g_{1.2}$: Page number design
- $g_{1.4}$: Comment in a document
- $g_{21.1}$: Changing initials in a comment
- $g_{21.2}$: Comment in a document
- $a_{1.2.1}$: Page number design
- $a_{1.2.2}$: Changing initials in a comment
- $a_{21.2.1}$: Changing initials in a comment
- $a_{21.2.2}$: Changing initials in a comment
- $a_{21.2.3}$: Changing initials in a comment

Actions:
- Execute
- Reset
- Override
- Append