# Lightly Supervised Learning of Procedural Dialog Systems

Svitlana Volkova

in collaboration with Bill Dolan, Pallavi Choudhury, Luke Zettlemoyer, Chris Quirk

# Outline

- Motivation
- Data
  - Query Logs
  - Office.com Help Pages
- Problem Formulation, Experimental Setup and Results
  - Task 1. Building **Dialog Trees** from Instructional Text
  - Task 2. Mapping Queries to **Initial Dialog States**
  - Task 3. Learning **Dialog State Updates** from Query Refinements
- Conclusions

# Intent-Oriented Dialog in Office

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."

# 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);

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

- Goal: building scalable dialogue systems in an automated way.
- Our approach:
  - learning procedural dialog systems from task-oriented textual resources in combination with light, non-expert supervision.

# Related Work

- **Grounded Language Learning**
  - Learning to understand instructions and complete the tasks automatically (Branavan et al., 2009, 2010; Kushman et al., 2009)
  - Understanding game strategy guides (Branavan et al., 2011)
  - Learning to follow navigational directions (Vogel and Jurafsky, 2010)
  - Learning from conversational interactions (Artzi and Zettlemoyer, 2011)
  - Learning to sportscast (Chen and Mooney, 2011)

- **Dialog Generation from Text**
  - Rule-based monologue to dialog generation (Paul Piwek et al. 2007)
  - Data-driven text to dialog generation (Piwek and Stoyanchev, 2010, 2011)

# Data: Web Query Logs

- Access to task knowledge in textual form:
  - examples of user intent statements (e.g., from search query logs);
  - instructional text (e.g., Office help pages).

- Where did we get query data?
  - Query Explore Platform (18-month log queries' click graph)
- Queries mapped to Office Help Pages
  - High frequency pain points (more than 1K queries per page)

# Data: Instructional Text

**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
- ✓ word 2007 insert page numbers not available
- ✓ how to insert page numbers on both side of the pages

Office | Search all of Office.com | bing

home | products | support | images | templates | downloads | more

Support / Word / Word 2007 Help and How-to / Page numbers

Add and format page numbers

**Topic**

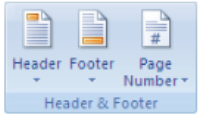
**Goal 1**

**Goal 2**

**Instructions**

Create a header or footer that contains only the page number

1. On the **Insert** tab, in the **Header & Footer** group, click **Page Number**.

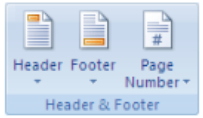



1. Click **Top of Page** or **Bottom of Page**, depending on where you want page numbers to appear in your document.
2. Choose a page number design from the gallery of designs.

NOTE If you don't see a gallery of page number designs, there might be a problem with the Building Blocks template on your computer. See I don't see galleries of page numbers, headers and footers, cover pages, or equations for information about how to fix this problem.

Place the page number in the side margin of the page

1. On the **Insert** tab, in the **Header & Footer** group, click **Page Number**.




1. Click **Page Margins**.
2. Choose a page number design from the gallery of designs.

NOTE If you don't see a gallery of page number designs, there might be a problem with the Building Blocks template on your computer. See I don't see galleries of page numbers, headers and footers, cover pages, or equations for information about how to fix this problem.

# Data Pros and Cons

**Pros**

- Tons of paraphrases for user intents
- Easy to get! Crowdsourcing! Direct query to help page mapping!

**Cons**

- 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

- Other domains:
  - recipes that describe how to cook meals, software help web pages that describe how to achieve goals - www.ehow.com or www.wikianswers.com

# Task 1. Building Dialog Trees from Instructions

I. Problem Formulation
II. User Action Type Classification Model
III. Experimental Setup and Results

# Problem Formulation

- Given:
  - help pages $p_1 \ldots p_n$ and queries $q_1 \ldots q_m$ per page $p_i$
- Find:
  - for every page $p_i$ extract a topic $t_i$ and goals $g_1 \ldots g_k$
  - for every goal $g_j$ in a topic $t_i$ extract instructions $i_1 \ldots i_l$

    classify user actions $a_u^1 \ldots a_u^l$ identify system actions $a_s^1 \ldots a_s^l$
  - construct dialog trees $f_1 \ldots f_k$

Office Help Pages (Topics) and Queries – $t_1 \ldots t_n$ and $q_1 \ldots q_m$

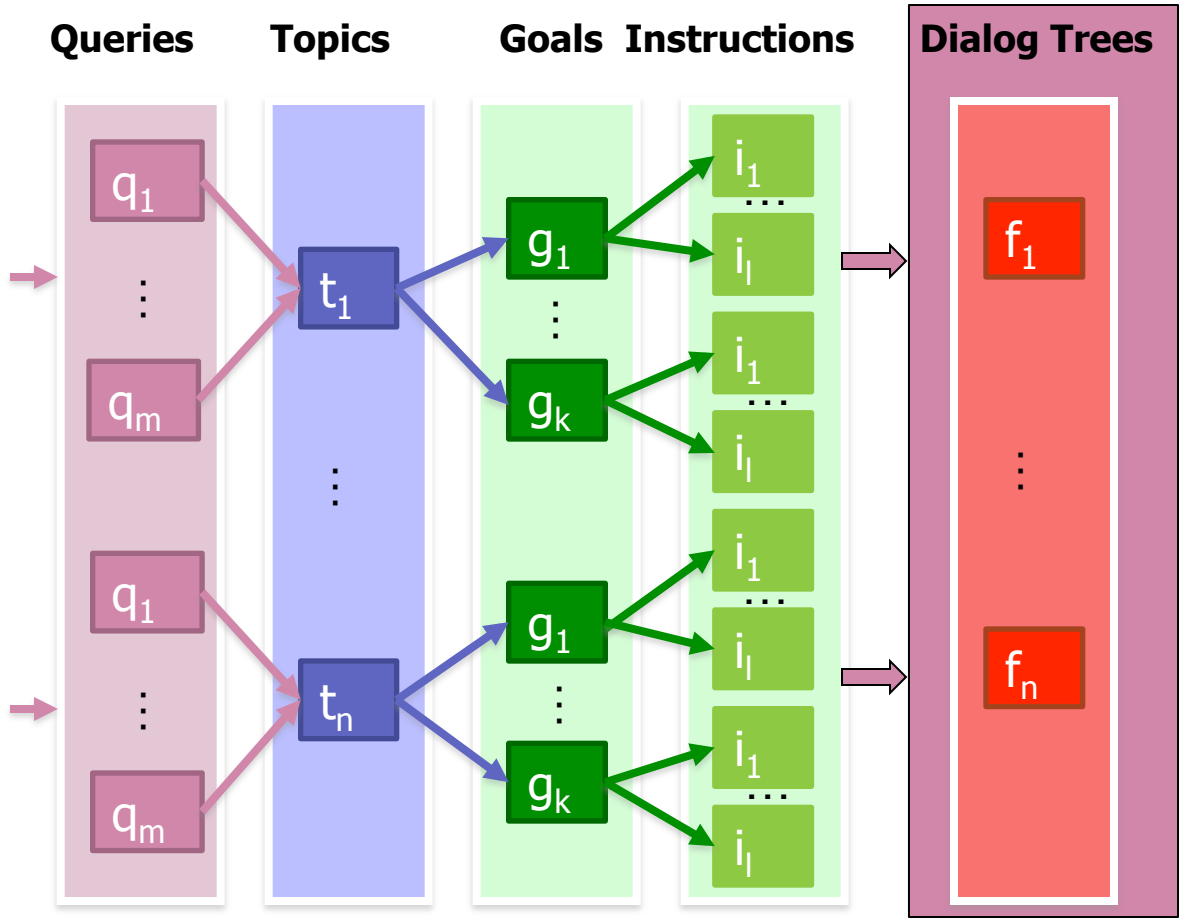
Goals and Instructions – $q_1 \ldots q_k$ and $i_1 \ldots i_l$

Trees – $f_1 \ldots f_k$

**Query Paraphrases**

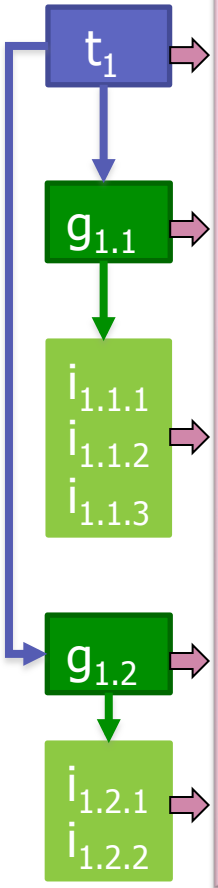
- *help with page numbers in the header*
- *how to number some pages*
- *adding page numbers*
- *how do you add numbering*

- *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?*



# Extracting Goals and Instructions

**Office Help Page**




## ADD AND FORMAT PAGE NUMBERS

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

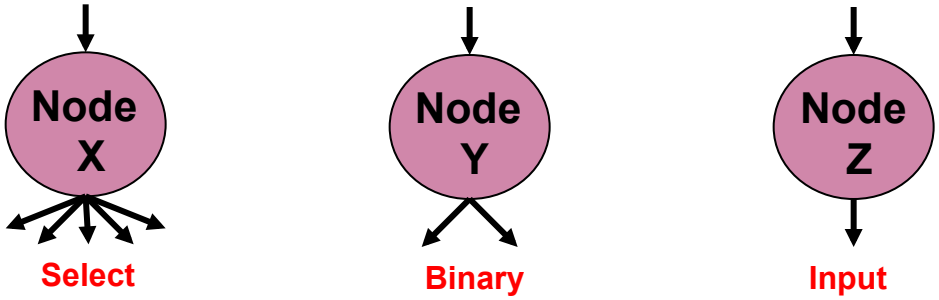
1. On the **Insert** tab, in the **Header & Footer** group, click **Page Number**.
2. Click **Top of Page** or **Bottom of Page**, depending on where you want page numbers to appear in your document.
3. Choose a page number design from the gallery of designs.

### 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 in your document.

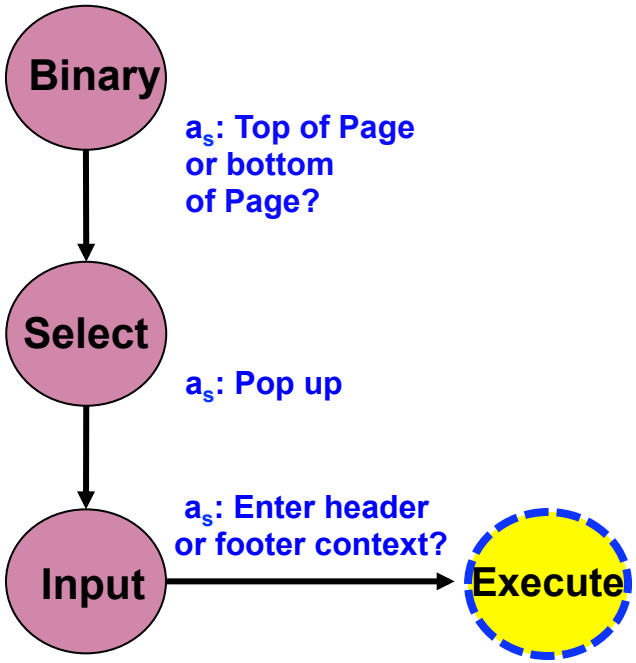
# User Actions

- Process instruction sets, identify steps requiring user action (different node types in a dialog tree represent user interaction points and interaction types):
  - **Selection** (picture, template, dialog box)
  - **Binary choice** (e.g. top or bottom of page)
  - **Input** (enter a comment, insert page number)
  - **None** (do not require user interaction)

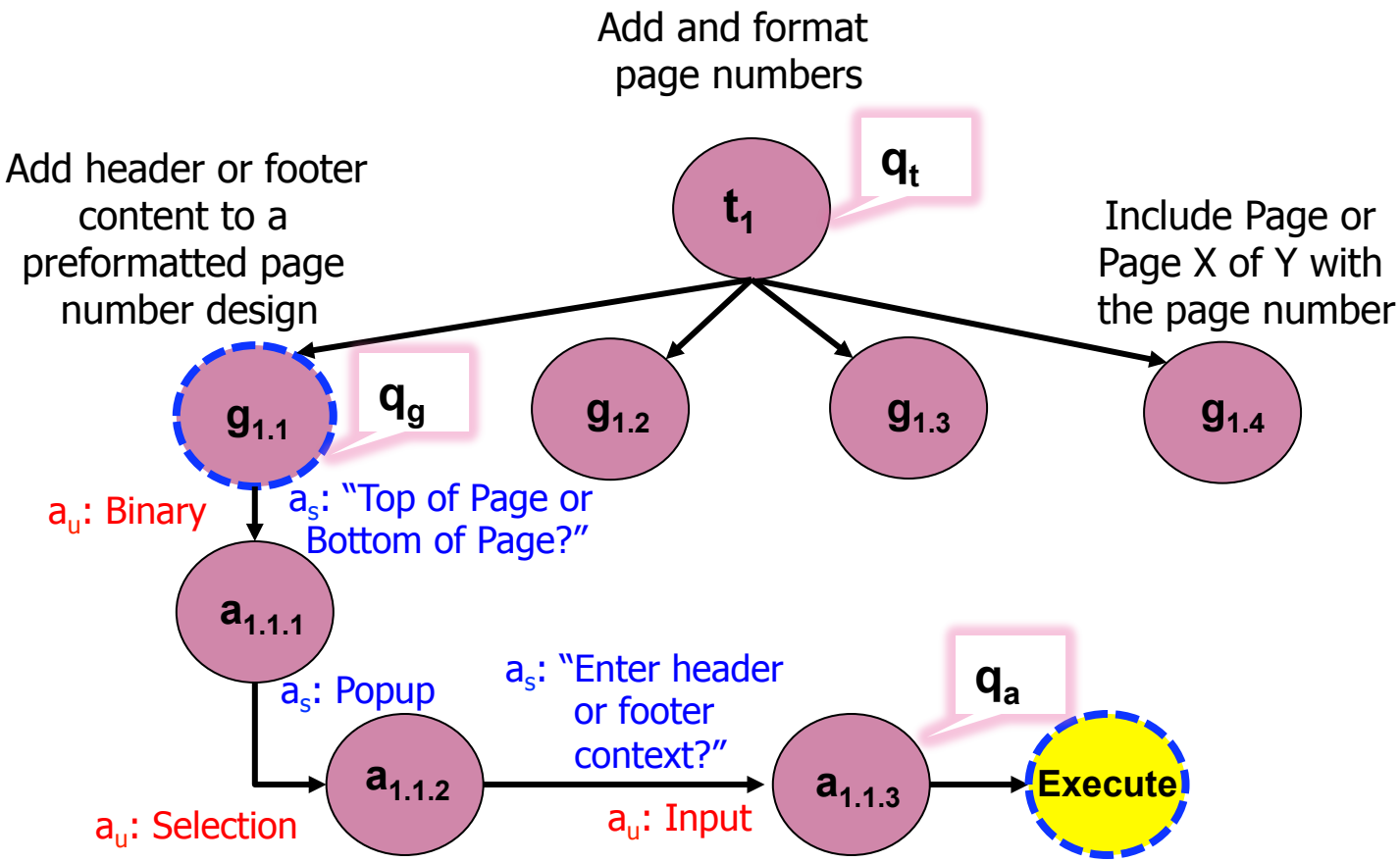


# System Actions

- Process instruction sets, identify steps requiring system actions:
  - **Ask a binary question**
    - "….Top Page or Bottom of Page, depending on where you want page numbers to appear in your document?"
  - **Display a popup**
    - "Choose a page number design from the gallery of designs"
  - **Display a prompt**
    - "Type the content, and then press TAB to position the content"
  - **Execute macro**



# Building Dialog Trees



# User Action Type Classification (1)

- Observe instructions $i_1 \ldots i_l$ with categories:
  - Binary, Selection, Input and None

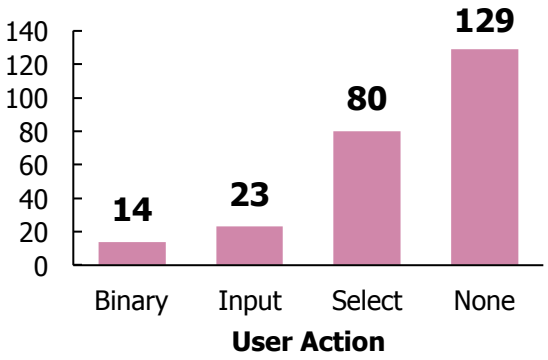
- Log-linear model (max. cond. likelihood)

$$p(a_u | i, \theta) = \frac{e^{\theta \cdot \phi(a_u, i)}}{\sum_{a'_u} e^{\theta \cdot \phi(a_u, i)}}$$

- Features are indicator functions of properties of the instructions and a particular class $\phi(a_u, i) \in \mathbb{R}^n$

# Experiment Setup

- Goal: classify user action types
- Data (manually labeled):
  - 30 topics with >1000 queries
    - pain points for Word users
  - 76 goals, 246 instructions
  - ~ 3 instructions per goal




- Training: 199 instructions
- Test: 47 instructions

# Experiment Results

| Feature Types | #Features | Accuracy |
|---|---|---|
| Baseline 1: Majority | - | 0.53 |
| Baseline 2: Heuristic | - | 0.64 |
| Ngrams | 10,556 | 0.89 |

Summary:

- Extracted topics, goals, instructions from Office Help Pages (OHP)
- OHP instructions can be easily classified into user actions with high accuracy → OHP are incredibly structured
- Build dialog trees automatically with limited annotation effort

# Task 2. Understanding Initial Queries

I. Problem Formulation
II. Initial Dialog State Classification Model
III. Experimental Setup and Results

# Problem Formulation

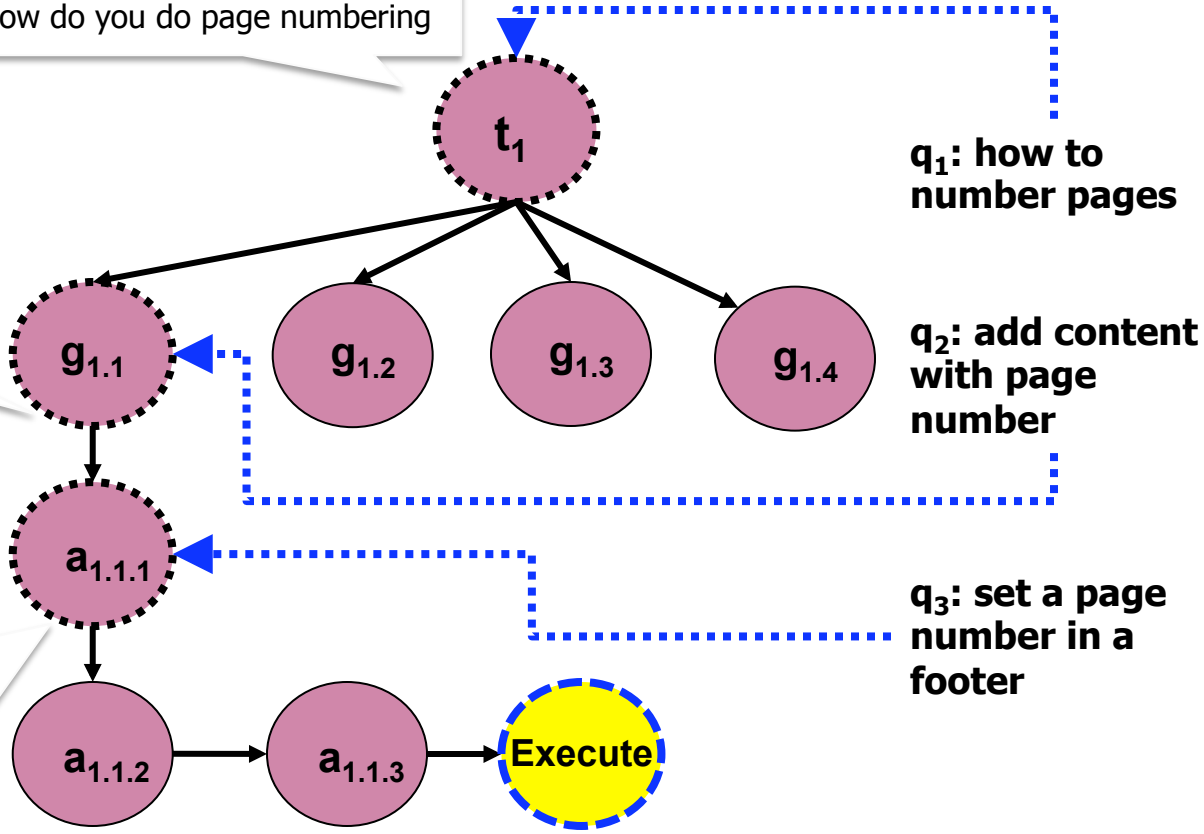
- Given:
  - Dialog trees $f_1 \ldots f_n$ with topics, goals and actions (includes annotated queries) and a new query q
- Find:
  - an initial dialog state $s_0$ that maximizes $p(s_0|q,\theta)$
  - predict the deepest relevant node in the dialog tree

# 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



# Initial Dialog State Classification (1)

- Observe a query q to node mapping which encodes an initial dialog state update $s_0$ with a category:

$$s_0 = [t_1, g_{1.1}, g_{1.2}, g_{1.2.1} \ldots, t_n, g_{n.1}, g_{n.1.1}]$$

- We employ a log-linear model to maximize initial dialog state distribution over the space of all nodes in a dialog network:

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

# Features

- **Lexical features:**
  - Query n-grams + stems + no stop words
- **State features:**
  - TFIDF 10 best scores
  - Query q overlap with system prompt $a_s$ from an initial state $s_0$
    - q: help with page numbers and $a_s$: choose a page number design
  - Query q overlap with queries from an initial state $s_0$ state
    - q: how to set a page number in a footer and $q_s$: how to insert page numbers in footer
  - Query q overlap with queries from state $s_0$ parents (history)

# Experiment Setup

- Goal: find initial dialog state $s_0$ given query q
- Data (manually mapped queries to nodes):
  - 5 - 10 queries per topic, goal
  - less queries for the action nodes
  - ~60% of action nodes do not have any queries
    - e.g., "Choose a page number design from the gallery of designs"

- Training: 838 query to initial state pairs
- Test: 134 query to initial state pairs

# Experiment Results (1)

| Feature Types | Topic | Goal | Action |
|---|---|---|---|
| Baseline 1: Random | 0.10 | 0.04 | 0.04 |
| Baseline 2: TFIDF 1 Best | 0.81 | 0.21 | 0.45 |
| Lexical (L) n-grams + stop words + stems | 0.92 | 0.66 | 0.63 |
| L + TFIDF 10Best | 0.94 | 0.66 | 0.64 |
| L + TFIDF 10Best + SystemPromptOverlap | 0.94 | 0.65 | 0.65 |
| L + TFIDF 10Best + QueryOverlap | 0.95 | 0.72 | 0.69 |
| All above + QueryHistoryOverlap | 0.96 | 0.73 | 0.71 |

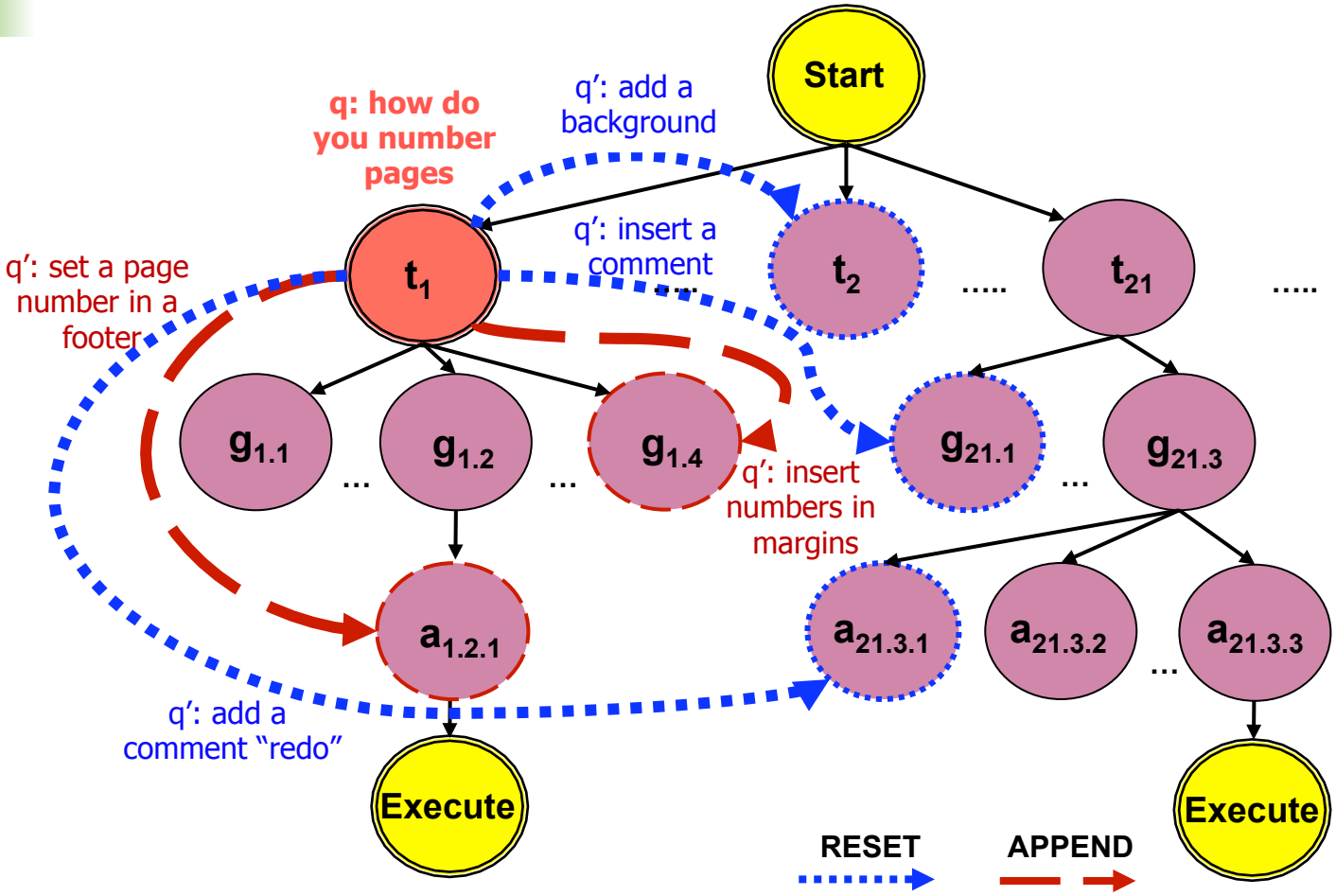
Summary:

- Even incorrect node assignments can lead to useful system performance! (as long as misclassification results being assigned to a too-high node within the correct dialog tree).

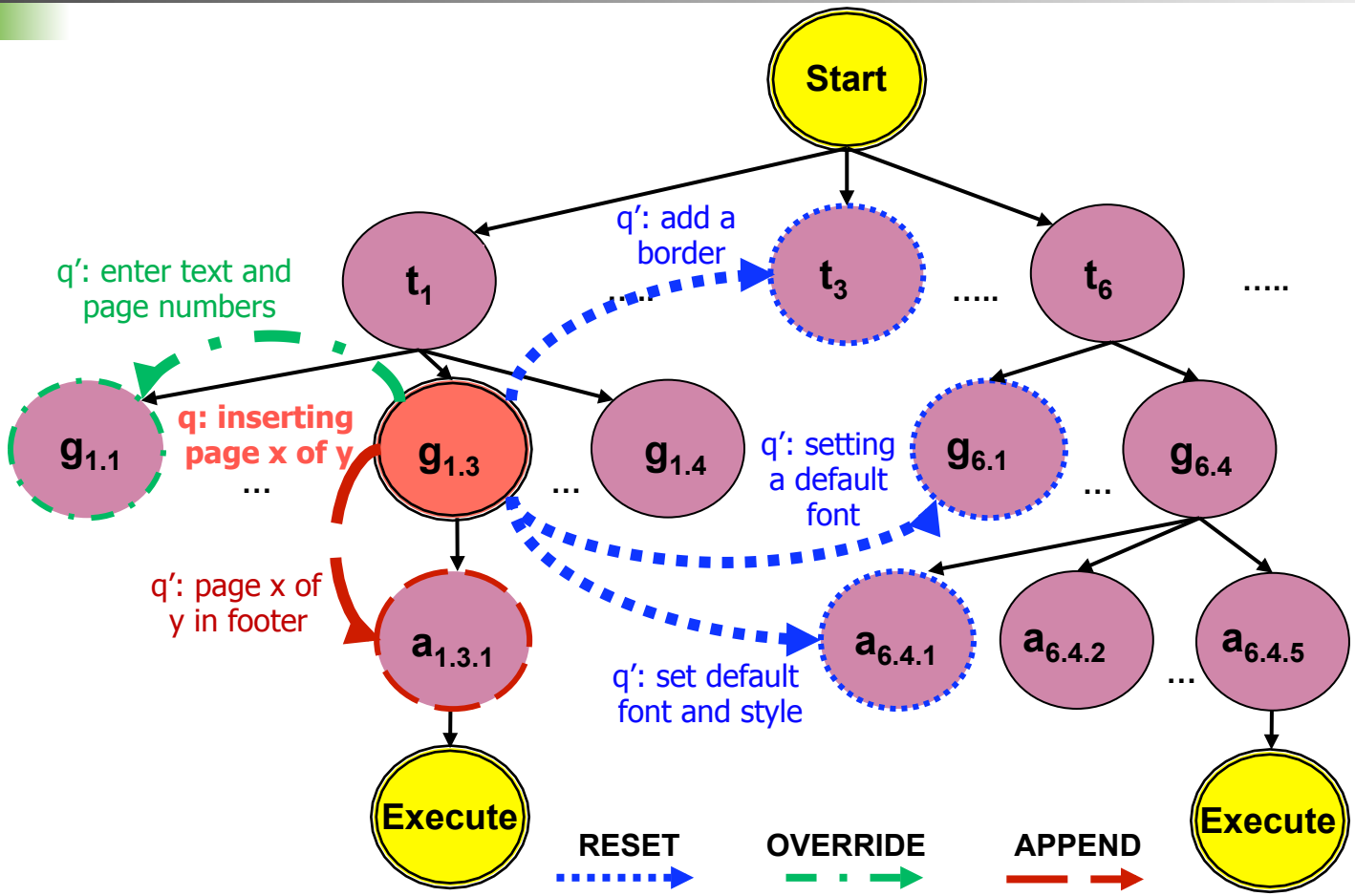
# Task 3. Understanding Query Refinements

I. Problem Formulation
II. Dialog State Update Classification Model
III. Experimental Setup and Results

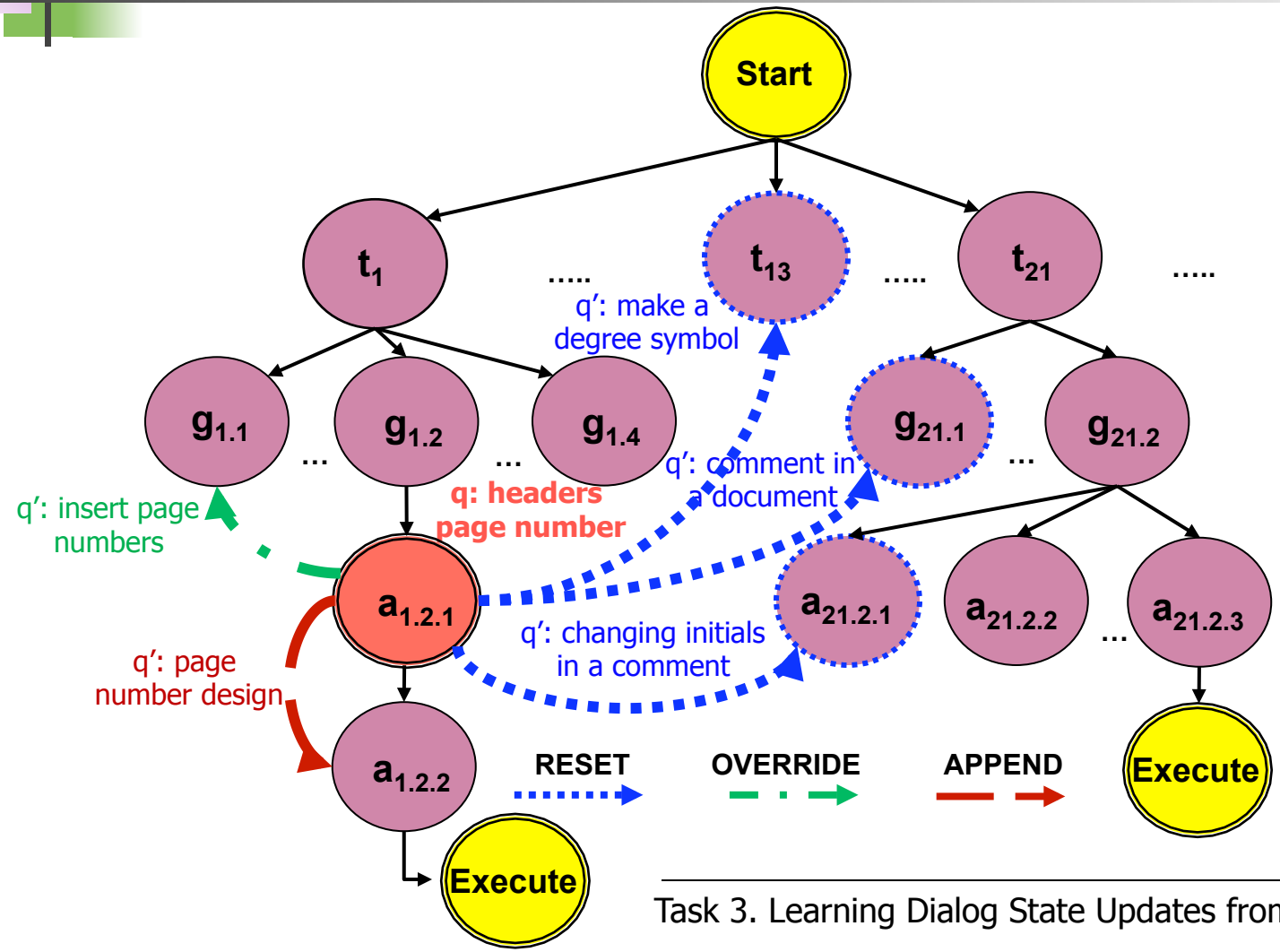
# Dialog State Update: From Topic Node



# Dialog State Update: From Goal Node



# Dialog State Update: From Action Node



# Problem Formulation

- Given a query q, system action $a_s$ from previous dialog state s and a new query q':
  - find a new dialog state s' at any level in the dialog tree that maximizes conditional probability:

$$p(s'|q', s, \theta) = p(s'|q', q, a_s, \theta).$$

# Dialog State Update Classification (1)

- Observe previous state s (includes prev. query q and system action $a_s$) and a new query q' pairs which encodes a new dialog state s' with a category:

$$[t_1, g_{1.1}, g_{1.2}, g_{1.2.1} \ldots, t_n, g_{n.1}, g_{n.1.1}]$$

- Log-linear model:

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

# Features (1)

- **Lexical features:**
  - Query n-grams + stems + no stop words

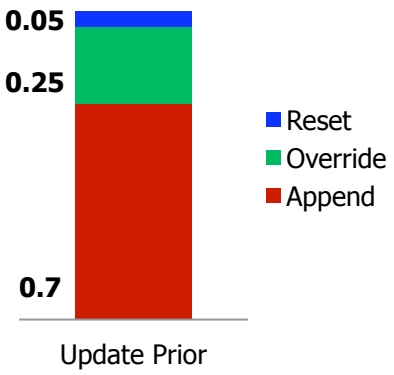
- **State features:**
  - TFIDF 10 best scores
  - Query q' overlap with prev. query q
    - q': how do add a comment and q: set a page number in footnote
  - Query q' overlap with system prompt $a_s$
    - q': make a degree symbol and $a_s$: choose a page number design

# Features (2)

- **Query and state overlap features:**
  - Query q′ overlap with queries from prev. state s
    - q′: footer page number and $q_s$: enter page x of y
  - Query q′ overlap with queries from new state s′
    - q′: footer page number and $q_{s'}$: how to insert page numbers in footer
- **Parent query overlap features:**
  - Query q′ overlap with queries from prev. state s parents
  - Query q′ overlap with queries from new. state s′ parents

# Experiment Setup

- Goal: predict a new state s′ given a new query q′ and prev. state s
- Simulate Dialog Turns
  - For every node/state n:
    - sample q ~ ($q_1^n$ … $q_m^n$)
    - get system action $a_s$
    - sample a new node n′ using prior probability
    - sample q′ ~ ($q_1^{n'}$ … $q_m^{n'}$)
- Training: 1100 prev. state s to new query q′ pairs
  Test: 440 prev. state s to new query q′ pairs



# Experiment Results

| Feature Types | Topic | Goal | Action | Append | Override |
|---|---|---|---|---|---|
| Lexical (L) | 0.92 | 0.76 | 0.78 | 0.90 | 0.89 |
| L + QOverlap | 0.93 | 0.80 | 0.80 | 0.92 | 0.83 |
| L + POverlap | 0.93 | 0.80 | 0.79 | 0.91 | 0.85 |
| L + QOverlap + POverlap | 0.94 | 0.80 | 0.80 | 0.93 | 0.85 |
| L + SQOverlap | 0.94 | 0.82 | 0.81 | 0.93 | 0.85 |
| L + S'QOverlap | 0.93 | 0.80 | 0.80 | 0.91 | 0.90 |
| L + S'QOverlap + ParentQOverlap | 0.94 | 0.80 | 0.80 | 0.91 | 0.86 |
| L + QOverlap + SQOverlap | 0.94 | 0.81 | 0.81 | 0.91 | 0.88 |
| L + SQOverlap + S'QOverlap | 0.95 | 0.84 | 0.83 | 0.94 | 0.88 |

# Small Scale User Study

- Select 8 from 76 goals, 48 interactions total
- Ask 6 users to rate the system on a scale from 1 to 5

| User Satisfaction | Av. Score ± Std. Dev. |
|---|---|
| Overall satisfaction | 3.42 ± 0.49 |
| Questions were relevant | 4.17 ± 0.41 |
| System managed ambiguity | 4.00 ± 0.89 |

| Objective Measures | Observed Rates |
|---|---|
| User achieved task | 83% |
| System asked irrelevant questions | 19% |
| System asked redundant questions | 18% |

# Conclusions

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

- Future Work:
  - Scale to new domains and 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 natural language

Thank you!

Questions