FALL 2020 CS 395T

TEXAS
The University of Texas at Austin

Translating Natural Language into Actions in Language Games

CS 395T: Topics in Natural Language Processing

Rohan Nair & Jiyang Zhang, The University of Texas at Austin

Overview

- "Language derives meaning from use."
  - Wittgenstein, 1953
- Wittgenstein's language games
  - Human wishes to accomplish task
  - Human can communicate with computer
  - Computer performs tasks as instructed by Human

Problem Definition

- High level formulation:
  - Define Computer set of actions A
  - Define Game states S
    - Current state $s_i \in S$ viewed by both Human and Computer
    - Human has a goal state $s_g \in S$
  - Human issues utterance L
  - Computer translates L into $a_t \in A$, game transitions from s_i to s_{i+1} using a transition function on (s_i, a_t).

Related Work

- Learning Language Games through Interaction
  - Block Stacking Game
  - Use semantic parsing model to translate
  - Wang et al. 2016

2 Case Studies:

- Executing Instructions in Situated Collaborative Interactions (Suhr et al. 2020)

- ChartDialogs: Plotting from Natural Language Instructions (Shao and Nakashole 2020)

TEXAS
The University of Texas at Austin

Executing Instructions in Situated Collaborative Interactions

Alane Suhr, Claudia Yan, Jacob Schluger, Stanley Yu, Hadi Khader, Marwa Mouallem, Iris Zhang, Yoav Artzi (EMNLP 2019)

CS 395T: Topics in Natural Language Processing

Jiyang Zhang, The University of Texas at Austin

Problem

- A collaborative scenario where a user not only instructs a system to complete tasks, but also acts alongside it.
- Learn to map user instructions to system actions.

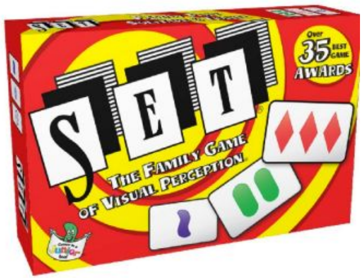
CerealBar

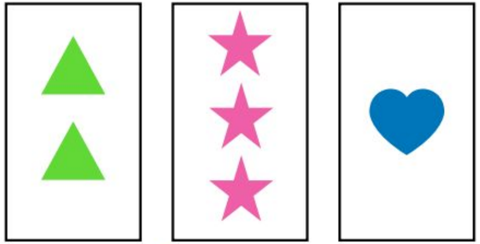
A situated collaborative game with sequential natural language instruction.

http://lil.nlp.cornell.edu/slides/2019_12_control_colab.pdf

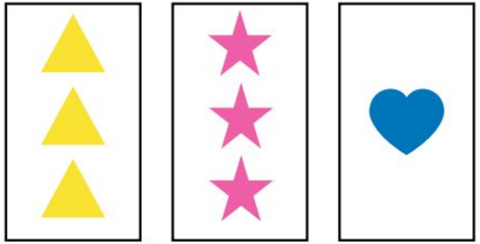
Game Objective

- Collect valid sets of three cards
- Valid: unique color, shape, and count
- Each set completed is one point





✓ Valid Set

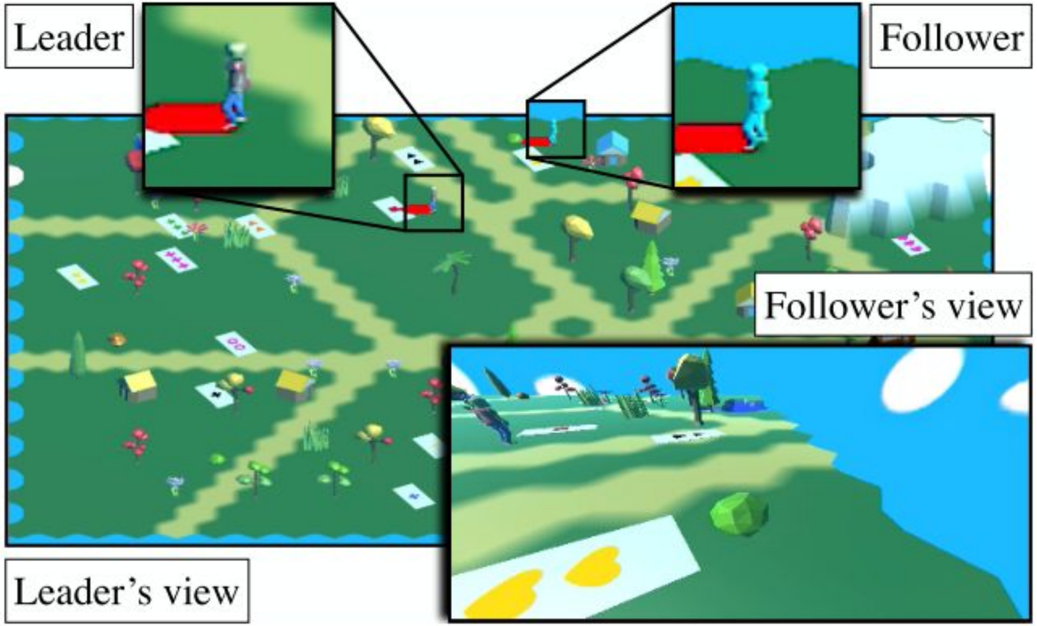


✗ Invalid Set

(two cards with three objects)

Collaboration (turn-based)

- The players select valid sets together
- The leader instructs follower using natural language
- Follower can not respond to the leader but execute instructions. They should not plan themselves.
- Incentivize collaboration:
  - **Observability**: leader sees complete board, follower only sees ahead
  - **Ability**: follower has more steps per trun



...

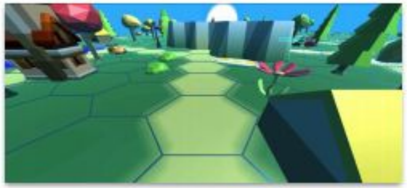
$\bar{x}_3$: *turn left and head toward the yellow hearts, but don’t pick them up yet. I’ll get the next card first.*

$\bar{x}_4$: ***Okay, pick up yellow hearts and run past me toward the bush sticking out, on the opposite side is 3 green stars***

`[Set made. New score: 4]`

...

Task

$f(\text{instruction},$  $,$  $) = \text{actions}$

- Context is history of interaction and structured observation from follower perspective
- Output discrete actions

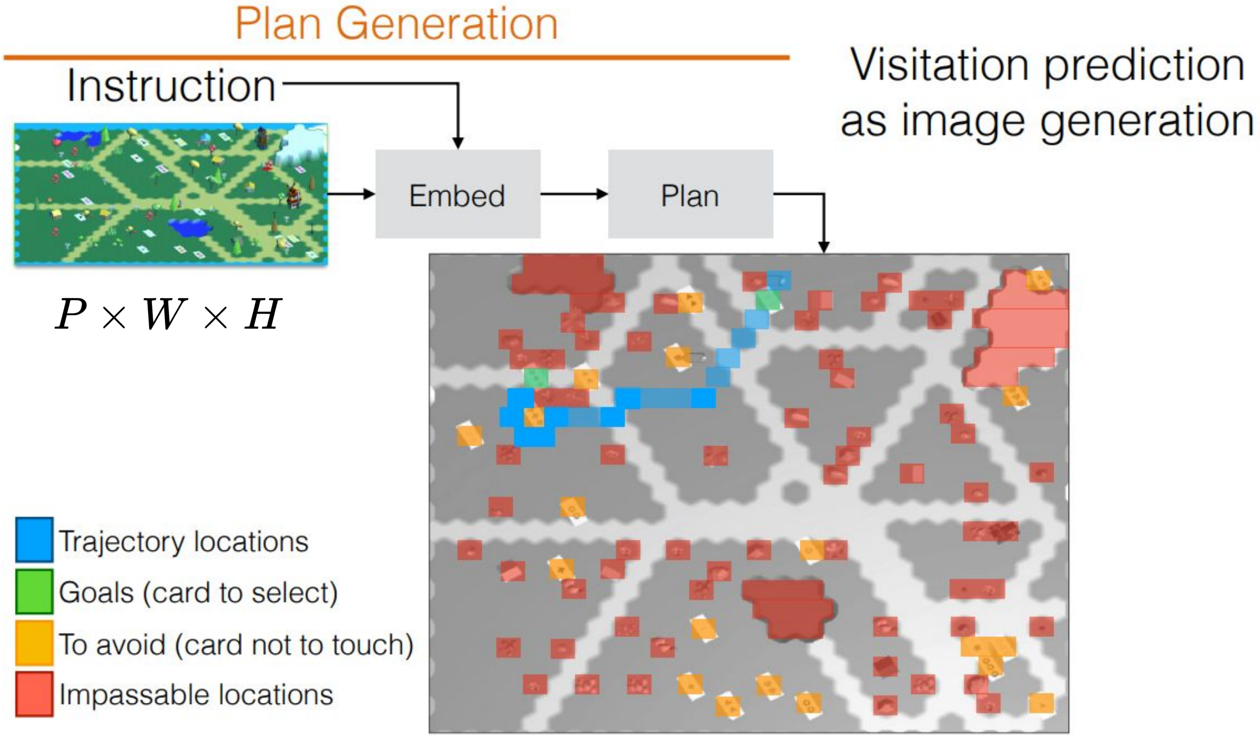
The CerealBar Scenario

- **Spatial reasoning:** interaction is in a dynamic 3D environment that keeps changing
- **Collaborative interaction:** working together is critical for success
- **Sequential instructions:** including dependencies, planning, changing goals, and sub-goals
- **User interaction:** the user continuously adapts and modifies their strategy

Model Overview

- Build on Visitation Prediction Model which casts planning as mapping instructions to the probability of visiting positions in the environment.
- Two extensions:
  - Plan distributions reason about obstacles and multiple goals
  - Recurrent action generation for more complex trajectories

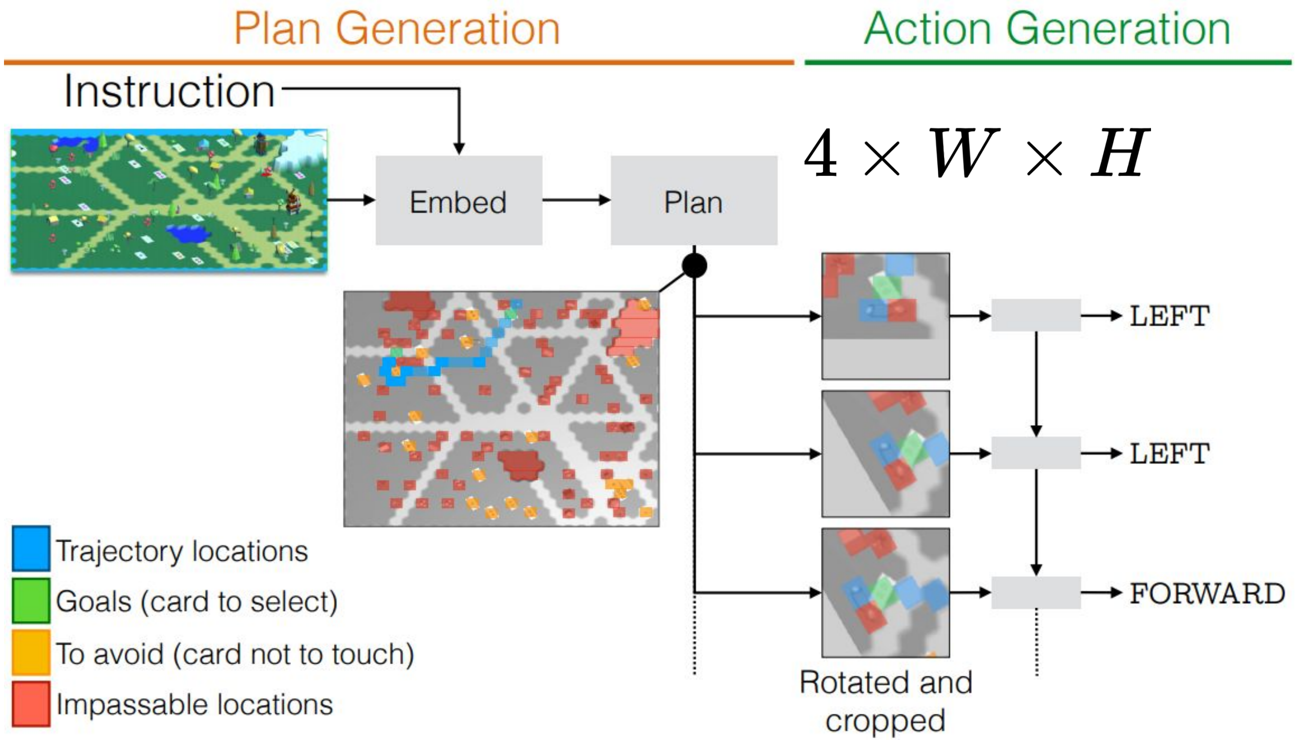
Model: first stage



Prediction distribution

- $p(\rho|s_t, \bar{x})$: the probability of visiting ρ while executing the instruction $\bar{x}$
- $p(GOAL = 1|\rho, s_t, \bar{x})$: the binary probability that ρ is a goal
- $p(AVOID = 1|\rho, s_t, \bar{x})$: the probability that agent must not pass in ρ
- $p(NOPASS = 1|\rho, s_t, \bar{x})$: the probability that agent cannot pass in ρ

Model: stage 2



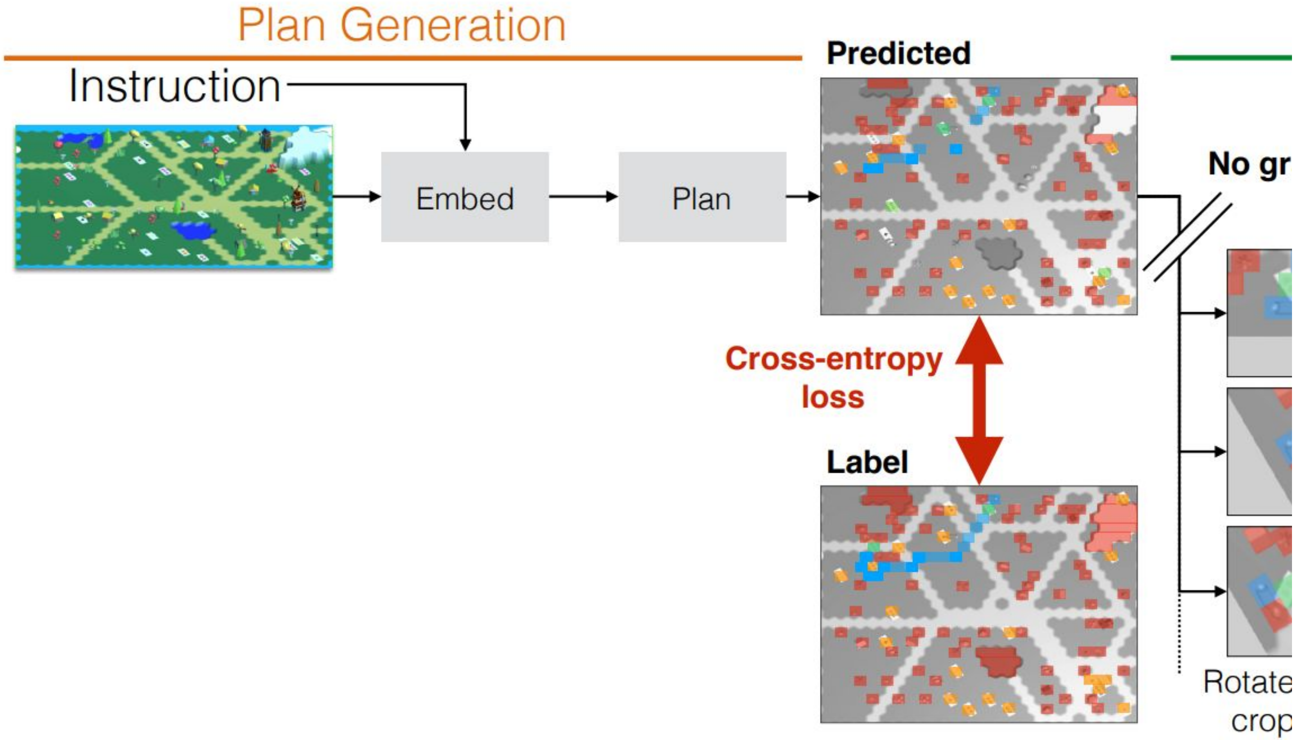
Training Data

- Recorded 1,202 successful human-human recorded games
- Each instruction is aligned with the sequence of actions the human follower performed

How to train

- Initialize both stages separately with supervised learning
- Train both stages of the model together
- Train to recover from error propagation between instructions

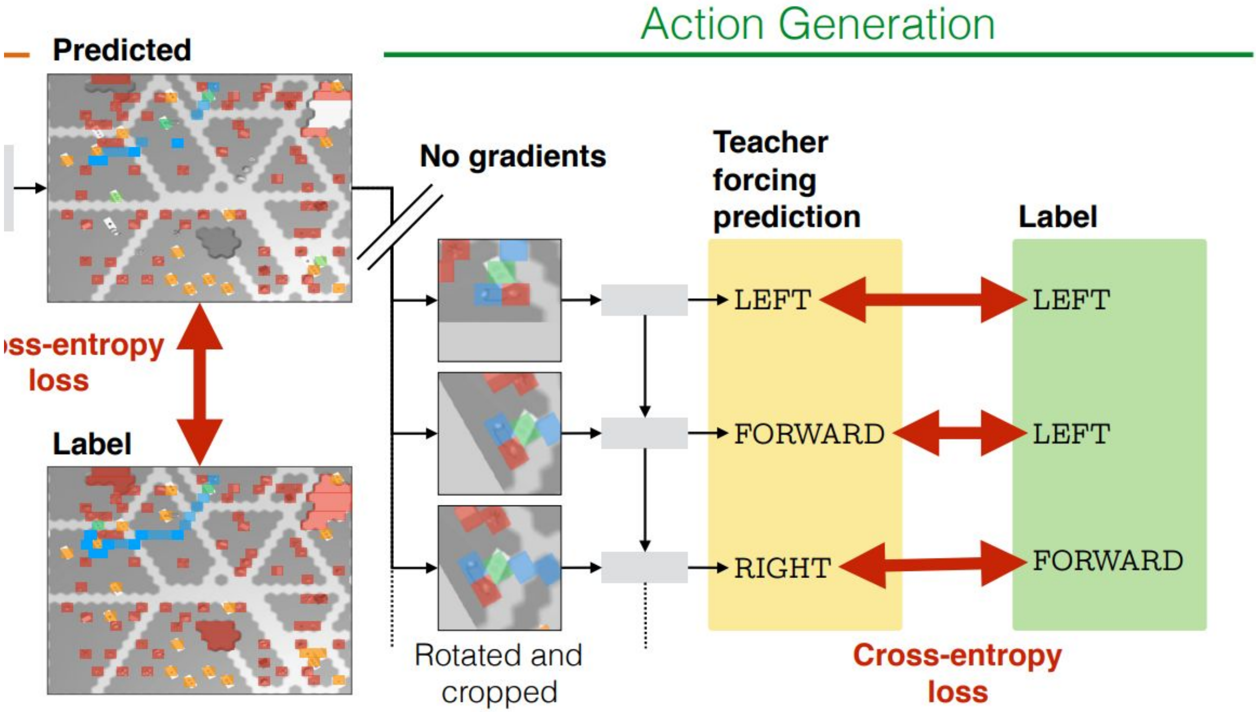
Learning: Plan Generation



Gold standard visitation distributions

- $p(\rho|s_t, \bar{x})$ label is proportion to number of states where the follower is in the position ρ
- $p(GOAL = 1|\rho, s_t, \bar{x})$ set the label to 1 for all ρ that contain a card that the follower changed its selection status during the interaction and 0 for all other.
- $p(AVOID = 1|\rho, s_t, \bar{x})$ label is 1 for all ρ that have cards that the follower does not change during interactions.
- $p(NOPASS = 1|\rho, s_t, \bar{x})$ label is 1 for all positions the agent cannot move onto.

Learning: Action Generation



Learning with error propagation

- Problems: there is no opportunity in the data to learn to recover from errors

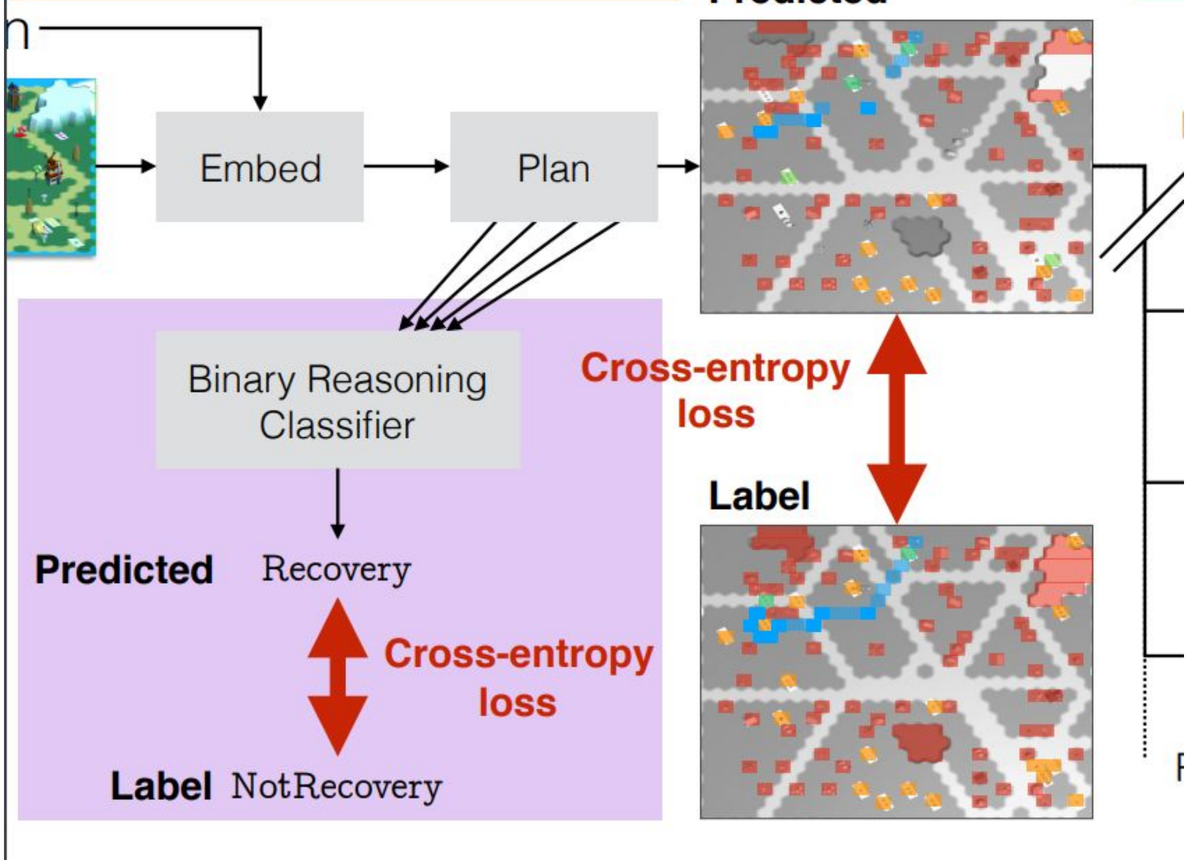
Augment the data with error recovery examples

- Run inference for each example using the current policy
- Compare the state *s* at the end of execution to the gold
- If the position or rotation of the agent are different, generate the shortest-path sequence of actions and add it to the recovery examples.

Optimize with Implicit action prediction

- The generated recovery examples may include sequences of state-action pairs that do not align with the original instruction.
- Identify such examples and treat them as requiring implicit actions (reasoning).
- All other examples are considered as not requiring implicit reasoning
- Train a classifier to determine whether the example requires implicit reasoning or not.

lan Generation



Cascaded Evaluation

- Instruction-level metrics ignore error propagation
- Instructions <1, 2, 3> -> <1, 2, 3> , < 2, 3> and <3>
- Proportion of the remaining instruction followed successfully
- Proportion of potential points scored

Systems

- Full Model
- SEQ2SEQ+ATTN
  - translates natural language instructions to action sequences based upon a representation of the observable world state.
- Static oracle that executes the gold actions

Metrics

- *Mean card state accuracy*: comparing the state of the cards after inference with the correct card state
- *Environment state accuracy*: comparing both cards and the agent's final position
- *Action sequence accuracy*: comparing the generated action sequence with the correct action sequence.
- *Full game points*
- *Mean proportion of instruction correctly executed*
- *Proportion of potential points scored*

Results

System	Card State Acc.	Env. State Acc.	Action Seq. Accuracy	Full Game Points	Prop. Instr. Followed	Prop. Points Scored
Development Results & Ablation Analysis						
Full model	58.2±0.5	32.6±0.8	15.8±0.5	0.66±0.1	20.5±1.2	18.1±0.8
– Trajectory distribution	38.5±2.7	10.1±2.7	5.5±2.6	0.29±0.02	10.0±0.9	7.9±0.7
– GOAL distribution	56.2±1.5	30.8±0.4	14.9±0.3	0.66±0.09	17.9±1.0	15.9±1.3
– AVOID distribution	57.0±0.3	32.6±1.6	15.4±1.3	0.63±0.04	18.8±1.5	17.8±0.7
– NOPASS distribution	59.2±0.5	32.0±0.8	15.0±0.5	0.70±0.03	18.4±0.9	16.6±0.9
– Action recurrence	42.3±1.5	16.7±1.2	10.0±0.7	0.42±0.03	12.8±1.7	10.7±0.5
– Fine-tuning	43.6±1.9	8.5±1.1	4.5±0.5	0.65±0.09	14.1±1.3	9.2±0.9
– Early goal auxiliary	57.2±2.3	31.2±1.7	14.9±1.6	0.65±0.05	17.9±1.1	16.5±0.7
– Example aggregation	59.4±1.8	32.0±1.0	15.7±0.6	0.65±0.09	20.4±1.4	16.5±0.4
– Implicit discriminator	57.5±2.1	32.7±1.0	16.4±0.3	0.70±0.02	18.8±1.8	16.7±0.6
– Instructions	15.5±1.5	2.7±1.5	1.2±1.2	0.24±0.07	4.4±1.0	4.6±0.7
+ Gold plan	87.4±0.5	80.2±0.2	63.4±0.2	–	–	–
SEQ2SEQ+ATTN	35.3±0.8	11.1±0.5	9.4±0.5	0.20±0.04	8.8±0.1	6.3±0.1
Static oracle	99.7	99.7	100.0	6.58	98.5	97.9
Test Results						
Full model	58.4	32.1	15.6	0.62	15.4	17.9
SEQ2SEQ+ATTN	37.3	10.8	8.5	0.22	8.7	6.5
Static oracle	99.7	99.7	100.0	6.66	96.8	95.6

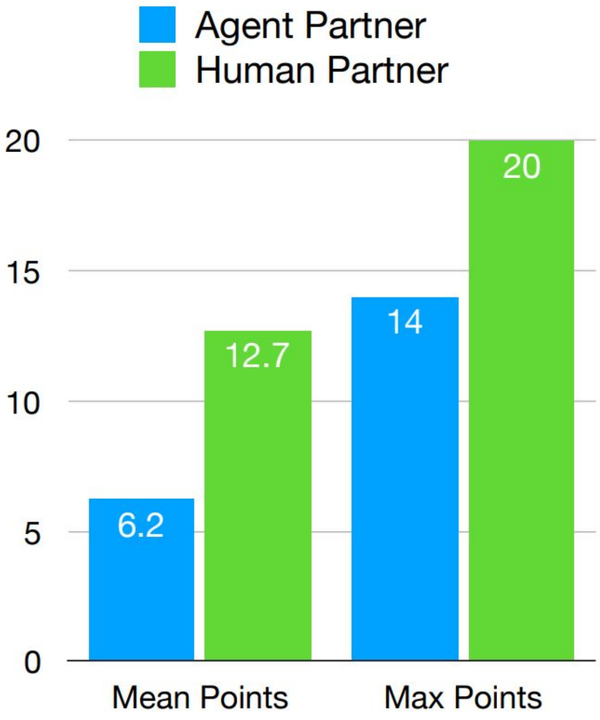
Table 1: Development and test results on all systems, including ablation results.

Ablation results

System	Card State Acc.	Env. State Acc.	Action Seq. Accuracy	Full Game Points	Prop. Instr. Followed	Prop. Points Scored
Development Results & Ablation Analysis						
Full model	58.2±0.5	32.6±0.8	15.8±0.5	0.66±0.1	20.5±1.2	18.1±0.8
– Trajectory distribution	38.5±2.7	10.1±2.7	5.5±2.6	0.29±0.02	10.0±0.9	7.9±0.7
– GOAL distribution	56.2±1.5	30.8±0.4	14.9±0.3	0.66±0.09	17.9±1.0	15.9±1.3
– AVOID distribution	57.0±0.3	32.6±1.6	15.4±1.3	0.63±0.04	18.8±1.5	17.8±0.7
– NOPASS distribution	59.2±0.5	32.0±0.8	15.0±0.5	0.70±0.03	18.4±0.9	16.6±0.9
– Action recurrence	42.3±1.5	16.7±1.2	10.0±0.7	0.42±0.03	12.8±1.7	10.7±0.5
– Fine-tuning	43.6±1.9	8.5±1.1	4.5±0.5	0.65±0.09	14.1±1.3	9.2±0.9
– Early goal auxiliary	57.2±2.3	31.2±1.7	14.9±1.6	0.65±0.05	17.9±1.1	16.5±0.7
– Example aggregation	59.4±1.8	32.0±1.0	15.7±0.6	0.65±0.09	20.4±1.4	16.5±0.4
– Implicit discriminator	57.5±2.1	32.7±1.0	16.4±0.3	0.70±0.02	18.8±1.8	16.7±0.6
– Instructions	15.5±1.5	2.7±1.5	1.2±1.2	0.24±0.07	4.4±1.0	4.6±0.7
+ Gold plan	87.4±0.5	80.2±0.2	63.4±0.2	–	–	–
SEQ2SEQ+ATTN	35.3±0.8	11.1±0.5	9.4±0.5	0.20±0.04	8.8±0.1	6.3±0.1
Static oracle	99.7	99.7	100.0	6.58	98.5	97.9
Test Results						
Full model	58.4	32.1	15.6	0.62	15.4	17.9
SEQ2SEQ+ATTN	37.3	10.8	8.5	0.22	8.7	6.5
Static oracle	99.7	99.7	100.0	6.66	96.8	95.6

Table 1: Development and test results on all systems, including ablation results.

Results: Live Games



- Our agent performs worse than humans
- However, followers adapt and can still use it effectively
  - Use simplified language, shorter instructions
  - Smaller vocab

Things I do not like

- Too many notations in the paper make their approach hard to understand
- Lack description on the model they built on (Visitation Prediction Network , LINGUNET)

Future work and discussion

- Remove full observability assumption
- Incorporating the interaction history to generate plan and implicitly reason
- Enable bidirectional communication to allow efficient collaboration between human and agent
- More complicated environment

FALL 2020 CS 395T

TEXAS
The University of Texas at Austin

ChartDialogs: Plotting from Natural Language Instructions

Yutong Shao and Ndapa Nakashole

CS 395T: Topics in Natural Language Processing

Rohan Nair, The University of Texas at Austin

Background

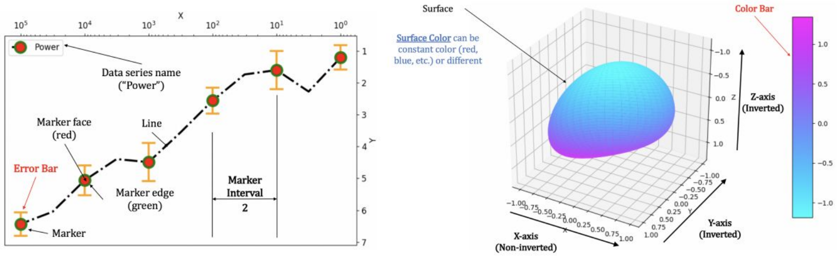
- Plotting is time consuming for novices
- 3 stages of plotting pipelines:
  - Describing the data
    - Functions: Pull Data, *Simple* Data Analysis
  - Describing the function
    - Functions: Specify function mathematically
  - Describing the plot
    - Functions: Manipulate the image output

Related Work

- Natural Language Interfaces (NLIs) studied in other parts of the pipeline
  - NLI + HCI works in describing the data
    - (Gao et al., 2015; Setlur et al., 2016; Srinivasan and Stasko, 2017; Yu and Silva, 2019; Sun et al., 2010).
  - NLI in describing the function
    - (*wolframalpha*)
- Plot manipulation is more structured than Conversational Image Editing, less complex than PL synthesis

Problem Statement

- Conversational plot updating agent
  - Describing a plot claimed as an iterative problem
- Slot filling goal-oriented dialog
  - Slots specific to plot type



(a) Line Chart

(b) 3D surface

Figure 1: Illustration of two of CHARTDIALOGS plot types. **(a) Line Chart** has slots such as *Line Style*. **(b)** A **3D Surface** has slots such as *Surface Color*.

Text Plot Specification

- Model plot spec as a key-value list (TP Spec)
- Definition (TP Spec):
  - Let S^t be the set of all relevant slots for a given plot type, t
  - For each slot $s_i \in S^t$, let the set of values it can take be V^t_i
  - $TP^t = \{(s_1 : v_1, s_2 : v_2, \ldots) : s_i \in S^t; v_i \in V^t_i\}$

Data Collection-Plot Generation

- One to One mapping from TP Spec to Plot image
- Randomly sample valid TP Specs to generate a corresponding plot image

Data Collection-Dialog Collection

- Wizard-of-Oz (WOZ) Collection Scheme over MTurk
  - One operator, one describer, shared working state
    - Plot state initialized blank, describer given goal state
  - Describer role:
    - issues command in natural language
  - Operator role:
    - Executes on natural language commands using operation panel
    - Can ask clarifying questions

Dataset Analysis – Size Statistics

- 3,284 Dialogs
- 15,754 Dialog Turns
- 141,876 tokens

	DSTC2 [2014] *(restaurant)*	SFX [2014] *(restaurant)*	WOZ2.0 [2017] *(restaurant)*	FRAMES [2017] *(travel)*	KVRET [2017] *(car)*	M2M* [2018] *(movie,rest)*	ImageEdits [2018] *(images)*	CHARTDIALOGS [2019] *(plots)*
# Dialogues	1,612	1,006	600	1,369	2,425	1,500	129	**3,284**
Total # turns	**23,354**	12,396	4,472	19,986	12,732	14,796	8,890	15,754
Total # tokens	199,431	108,975	50,264	**251,867**	102,077	121,977	59,653	141,876
Avg. turns per dialo.	14.49	12.32	7.45	**14.60**	5.25	9.86	unk	4.80
Avg. tokens per turn	8.54	8.79	11.24	**12.60**	8.02	8.24	unk	9.01
Total unique tokens	986	1,473	2,142	**12,043**	2,842	1,008	2,299	2,652
# Slots	8	14	4	**61**	13	14	unk	53
# Values	212	1847	99	3871	1363	138	unk	328

Table 1: Comparison of CHARTDIALOGS to other single domain goal-oriented dialog data sets. *M2M is largely on the restaurant domain but also includes movies

Dataset Analysis – GPT-2 Statistics

- Lower half of GPT-2 Perplexity Scores
  - Distribution has long tail
  - Average Perplexity: 77,188.58 (whole), 399.78 (lower half)

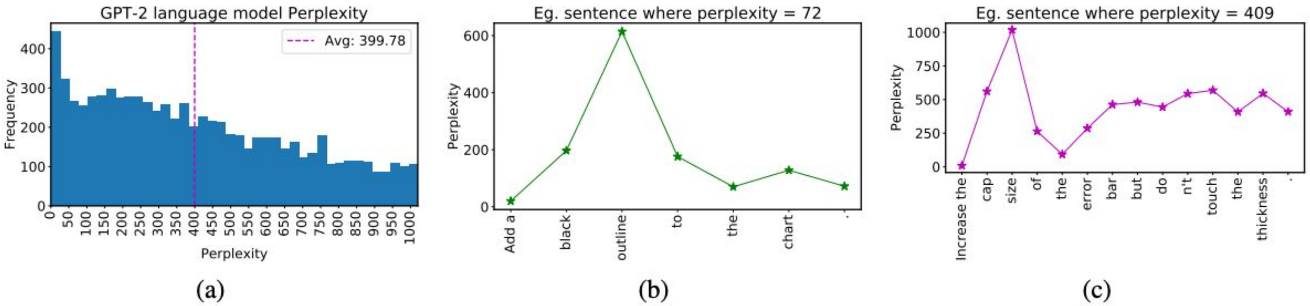


Figure 2: **(a)**: Distribution of perplexity of the utterances. **(b) and (c)**: average per word surprise of a growing sentence as new words are added to the sentence. High perplexity is a result of plot-specific terms line ‘outline’, and ‘cap’ arising in unexpected contexts.

Dataset Analysis – Turn Analysis

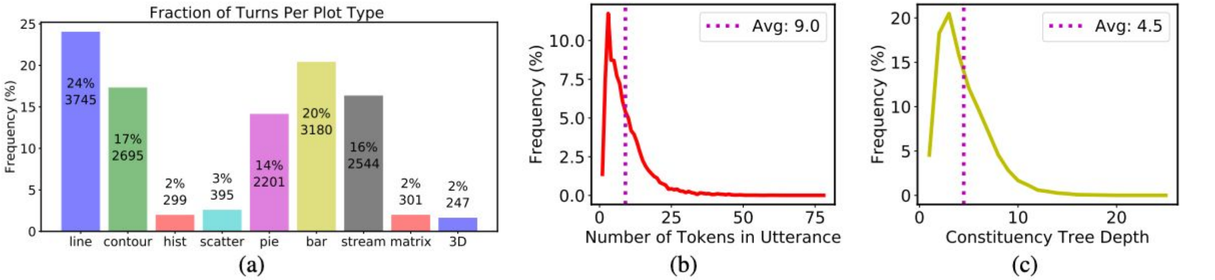
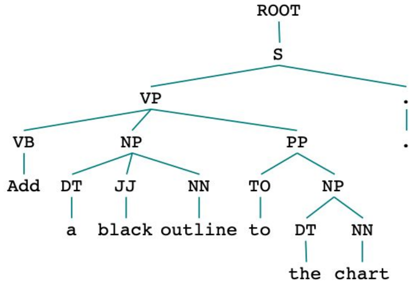
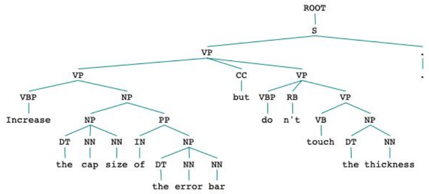


Figure 3: **(a)** Dialog turns per CHARTDIALOGS plot type. Distributions of **(b)** Words per utterance, and **(c)** Constituency tree depth for utterances.



(a) Constituency tree depth = 4



(b) Constituency tree depth = 8

Figure 4: Two CHARTDIALOGS utterances with different constituency tree depths. The average tree depth in the dataset is 4.5.

Baseline Methods

- Utilize seq2seq with Encoder/Decoder Stack + attention
- Input/Output Formulation
  - Input
    - 3 sources: current TP Spec, current plot as image, dialog history
  - Output
    - ΔTPSpec, either sequence or predicted using sequence of classifiers
- Decoder Output Probability
  - $$p(\mathbf{y}) = \prod_{t=1}^{T} p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c_t^*)$$

Baseline Methods – Sequence Output

- S2S-Plot + TXT
- S2S-TXT
- S2S-NoState
- S2S-NoUtterance

Baseline Methods – Sequential MLP

- MaxEnt
- RNN+MLP
- Transformer+MLP

Results: Token Granularity

- PAIR:
  - Concat slot name + value (“slot name:slot value”)
- SINGLE:
  - Split slot name + value (2 predictions)
- SPLIT:
  - Slot names and values split into words
  - “x_axis_scale:log” –> “x” + “axis” + “scale” + “:” + “log”

Results-Quantitative Analysis

- Training: 2,628, Validation: 328, Test: 329 dialogs
- Training: 11,903 Validation:1,562, Test: 1,481 datapoints

Methods	SPLIT	SINGLE	PAIR
S2S-PLOT+TXT	0.585	**0.613**	0.594
S2S-TXT	0.601	**0.613**	0.591
S2S-NoState	0.525	0.549	0.535
S2S-NoUtterance	0.060	0.047	0.046
MaxEnt	0.196	0.265	0.422
RNN+MLP	0.328	0.324	0.325
Transformer+MLP	0.311	n/a[6]	n/a[6]

Table 2: Exact match plotting performance.

Methods	SPLIT	SINGLE	PAIR
S2S-PLOT+TXT	0.871	**0.890**	0.888
S2S-TXT	0.874	**0.893**	0.885
S2S-NoState	0.847	0.866	0.863
S2S-NoUtterance	0.316	0.306	0.155
MaxEnt	0.677	0.734	0.806
RNN+MLP	0.714	0.712	0.724
Transformer+MLP	0.723	n/a[6]	n/a[6]

Table 3: Slot change F1 plotting performance.

Results – Plot Breakdown

Plot type	S2S-TXT		S2S-PLOT+TXT	
	Exact Match	Slot F1	Exact Match	Slot F1
Line	0.602±0.026	0.889±0.005	0.605±0.011	0.888±0.006
Bar	0.572±0.022	0.873±0.004	0.565±0.020	0.866±0.004
Pie	0.685±0.009	0.896±0.005	0.691±0.005	0.894±0.008
Contour	0.618±0.006	0.916±0.004	0.624±0.015	0.913±0.004
Streamline	0.610±0.016	0.901±0.009	0.598±0.023	0.895±0.007
Histogram	0.476±0.048	0.886±0.007	0.505±0.026	0.890±0.019
Scatter	0.492±0.026	0.849±0.014	0.492±0.017	0.851±0.014
Matrix	0.717±0.022	0.944±0.006	0.683±0.033	0.939±0.004
3D Surface	0.733±0.047	0.910±0.023	0.768±0.041	0.928±0.023
Total	0.613±0.005	0.893±0.002	0.613±0.005	0.890±0.002

Table 4: Exact match and Slot F1 score by plot type, under the SINGLE granularity.

Results-Qualitative Analysis

- Validate performance of original MTurk Operator over 444 Partial Dialogs
- 3 MTurk operators given partial dialog, asked to produce operation
- Cohen's Kappa: .889
  - Between Original MTurk worker and Majority of 3 partial dialog MTurk Operators

Original	New	Proportion
$\surd$	$\surd\surd\surd$	55.1%
$\surd$	$\surd\surd\times$	17.5%
$\surd$	$\surd\times\times$	2.4%
$\surd$	$\times\times\times$	0.0%
$\times$	$\surd\surd\surd$	8.0%
$\times$	$\surd\surd\times$	10.3%
$\times$	$\surd\times\times$	4.4%
$\times$	$\times\times\times$	2.3%
Total		100.0%

Table 5: Agreement evaluation result. $\surd$ stands for "exact match with majority" and $\times$ for "no exact match with majority". The majority is obtained slot-wise, i.e. the majority for each slot is obtained separately.

Error Analysis

- Human performance: 76.8% EM
  - Subset of 180 Examples
  - Top model: 61.3% EM
- 2 Ambiguity Error Classes:
  - Unspecified new Plot
  - Ambiguous Value
- Human Errors
  - Operator overlooks Describer

Previous State	(no grid lines)
Dialog History	invert y axis , red dashed **gridlines** , markers should be down triangle
Gold Output	grid_line_type horizontal
Model Output	grid_line_type both

(a) Ambiguity: unspecified new slot

Previous State	font_size large
Dialog History	make font size **smaller** again , sorry
Gold Output	font_size medium
Model Output	font_size small

(b) Ambiguity: ambiguous value

Table 6: Examples of different kinds of ambiguities.

Error Analysis – Model Errors

Previous State	line_style dotted
Dialog History	[Desc][7] dot line dot line [Op] this is dot , do you mean dot-dash ? [Desc] that would be it ... sorry
Gold Output	line_style dashed_dots
Model Output	line_style dotted

(a) Error: multi-turn dialog history

Previous State	(empty plot)
Dialog History	matrix display , **yellow to red** , x axis inverted on top , y axis inverted on right
Gold Output	color_map transparent_yellow_to_solid_red
Model Output	color_map red_to_yellow

(b) Error: complex slot value

Previous State	(empty plot)
Dialog History	hello , we have a bar plot ... orange bars with a black outline , **log style** please
Gold Output	y_axis_scale log
Model Output	y_axis_scale linear

(c) Error: infrequent expression

Table 7: Examples of different kinds of model errors.

Discussion Points

- The authors introduce examples of StackOverflow questions tagged with matplotlib. Is there a way to assess performance on this real world use case?
- Models built on this dataset will struggle with things like new features and patching. Do systematic ways to collect new training data seem reasonable?

YOU
BELONG
HERE

CNS Diversity and Inclusion • cns.utexas.edu/diversity

Questions?