Learning Symmetric Collaborative Dialogue Agents with Dynamic Knowledge Graph Embeddings

He He and **Anusha Balakrishnan** and **Mihail Eric** and **Percy Liang**
Computer Science Department, Stanford University
{hehe,anusha28,meric,pliang}@cs.stanford.edu

Abstract

We study a *symmetric collaborative dialogue* setting in which two agents, each with private knowledge, must strategically communicate to achieve a common goal. The open-ended dialogue state in this setting poses new challenges for existing dialogue systems. We collected a dataset of 11K human-human dialogues, which exhibits interesting lexical, semantic, and strategic elements. To model both structured knowledge and unstructured language, we propose a neural model with dynamic knowledge graph embeddings that evolve as the dialogue progresses. Automatic and human evaluations show that our model is both more effective at achieving the goal and more human-like than baseline neural and rule-based models.

1 Introduction

Current task-oriented dialogue systems (Young et al., 2013; Wen et al., 2017; Dhingra et al., 2017) require a pre-defined dialogue state (e.g., slots such as food type and price range for a restaurant searching task) and a fixed set of dialogue acts (e.g., request, inform). However, human conversation often requires richer dialogue states and more nuanced, pragmatic dialogue acts. Recent open-domain chat systems (Shang et al., 2015; Serban et al., 2015b; Sordoni et al., 2015; Li et al., 2016a; Lowe et al., 2017; Mei et al., 2017) learn a mapping directly from previous utterances to the next utterance. While these models capture open-ended aspects of dialogue, the lack of structured dialogue state prevents them from being directly applied to settings that require interfacing with structured knowledge.

In order to bridge the gap between the two types of systems, we focus on a *symmetric collaborative dialogue* setting, which is task-oriented but encourages open-ended dialogue acts. In our setting, two agents, each with a private list of items with attributes, must communicate to identify the unique shared item. Consider the dialogue in Figure 1, in which two people are trying to find their mutual friend. By asking "do you have anyone who went to columbia?", B is suggesting that she has some Columbia friends, and that they probably work at Google. Such conversational implicature is lost when interpreting the utterance as simply an information request. In addition, it is hard to define a structured state that captures the diverse semantics in many utterances (e.g., defining "most of", "might be"; see details in Table 1).

To model both structured and open-ended context, we propose the *Dynamic Knowledge Graph Network* (DynoNet), in which the dialogue state is modeled as a knowledge graph with an embedding

Friends of agent A:

Name	School	Major	Company
Jessica	Columbia	Computer Science	Google
Josh	Columbia	Linguistics	Google
...	...	...	...

A: Hi! Most of my friends work for Google
B: do you have anyone who went to columbia?
A: *Hello?*
A: I have Jessica a friend of mine
A: and Josh, both went to columbia
B: *or anyone working at apple?*
B: SELECT (Jessica, Columbia, Computer Science, Google)
A: SELECT (Jessica, Columbia, Computer Science, Google)

Figure 1: An example dialogue from the MutualFriends task in which two agents, A and B, each given a private list of a friends, try to identify their mutual friend. Our objective is to build an agent that can perform the task with a human. Crosstalk (Section 2.3) is *italicized*.

for each node (Section 3). Our model is similar to EntNet (Henaff et al., 2017) in that node/entity embeddings are updated recurrently given new utterances. The difference is that we structure entities as a knowledge graph; as the dialogue proceeds, new nodes are added and new context is propagated on the graph. An attention-based mechanism (Bahdanau et al., 2015) over the node embeddings drives generation of new utterances. Our model's use of knowledge graphs captures the grounding capability of classic task-oriented systems and the graph embedding provides the representational flexibility of neural models.

The naturalness of communication in the symmetric collaborative setting enables large-scale data collection: We were able to crowdsource around 11K human-human dialogues on Amazon Mechanical Turk (AMT) in less than 15 hours.[1] We show that the new dataset calls for more flexible representations beyond fully-structured states (Section 2.2).

In addition to conducting the third-party human evaluation adopted by most work (Liu et al., 2016; Li et al., 2016b,c), we also conduct partner evaluation (Wen et al., 2017) where AMT workers rate their conversational partners (other workers or our models) based on fluency, correctness, cooperation, and human-likeness. We compare DynoNet with baseline neural models and a strong rule-based system. The results show that DynoNet can perform the task with humans efficiently and naturally; it also captures some strategic aspects of human-human dialogues.

The contributions of this work are: (i) a new symmetric collaborative dialogue setting and a large dialogue corpus that pushes the boundaries of existing dialogue systems; (ii) DynoNet, which integrates semantically rich utterances with structured knowledge to represent open-ended dialogue states; (iii) multiple automatic metrics based on bot-bot chat and a comparison of third-party and partner evaluation.

2 Symmetric Collaborative Dialogue

We begin by introducing a collaborative task between two agents and describe the human-human dialogue collection process. We show that our data exhibits diverse, interesting language phenomena.

[1]The dataset is available publicly at https://stanfordnlp.github.io/cocoa/.

2.1 Task Definition

In the symmetric collaborative dialogue setting, there are two agents, A and B, each with a private knowledge base—KB_A and KB_B, respectively. Each knowledge base includes a list of *items*, where each item has a value for each *attribute*. For example, in the MutualFriends setting, Figure 1, items are friends and attributes are name, school, etc. There is a shared item that A and B both have; their goal is to converse with each other to determine the shared item and select it. Formally, an agent is a mapping from its private KB and the dialogue thus far (sequence of utterances) to the next utterance to generate or a selection. A dialogue is considered *successful* when both agents correctly select the shared item. This setting has parallels in human-computer collaboration where each agent has complementary expertise.

2.2 Data collection

We created a schema with 7 attributes and approximately 3K entities (attribute values). To elicit linguistic and strategic variants, we generate a random scenario for each task by varying the number of items (5 to 12), the number attributes (3 or 4), and the distribution of values for each attribute (skewed to uniform). See Appendix A and B for details of schema and scenario generation.

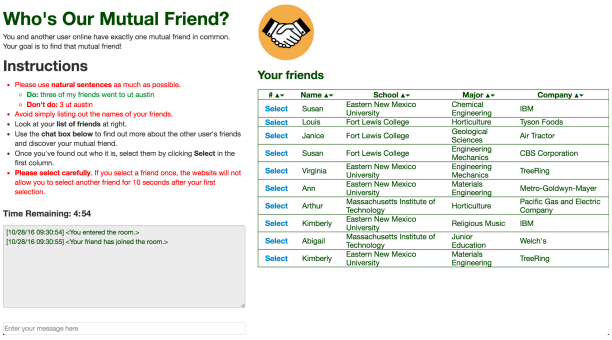

Figure 2: Screenshot of the chat interface.

We crowdsourced dialogues on AMT by randomly pairing up workers to perform the task within 5 minutes.[2] Our chat interface is shown in Figure 2. To discourage random guessing, we prevent workers from selecting more than once every 10 seconds. Our task was very popular and we col-

[2]If the workers exceed the time limit, the dialogue is marked as unsuccessful (but still logged).

Type	%	Easy example	Hard example
Inform	30.4	I know a judy. / I have someone who studied the bible in the afternoon.	**About equal** indoor and outdoor friends / **me too**. his major is forestry / **might be** kelly
Ask	17.7	Do any of them like Poi? / What does your henry do?	What can you tell me about our friend? / **Or maybe** north park college?
Answer	7.4	None of mine did / Yup / They do. / Same here.	yes 3 of them / No he likes poi / yes if boston college

Table 1: Main utterance types and examples. We show both standard utterances whose meaning can be represented by simple logical forms (e.g., ask(`indoor`)), and open-ended ones which require more complex logical forms (difficult parts in bold). Text spans corresponding to entities are underlined.

Phenomenon	Example
Coreference	(I know one Debra) does she like the indoors? / (I have two friends named TIffany) at World airways?
Coordination	keep on going with the fashion / Ok. let's try something else. / go by hobby / great. select him. thanks!
Chit-chat	Yes, that is **good ole** Terry. / All indoorsers! **my friends hate nature**
Categorization	same, most of mine are **female too / Does any of them names start with B**
Correction	I know one friend into Embroidery - her name is Emily. **Sorry – Embroidery friend is named Michelle**

Table 2: Communication phenomena in the dataset. Evident parts is in bold and text spans corresponding to an entity are underlined. For coreference, the antecedent is in parentheses.

lected 11K dialogues over a period of 13.5 hours.[3] Of these, over 9K dialogues are successful. Unsuccessful dialogues are usually the result of either worker leaving the chat prematurely.

2.3 Dataset statistics

We show the basic statistics of our dataset in Table 3. An utterance is defined as a message sent by one of the agents. The average utterance length is short due to the informality of the chat, however, an agent usually sends multiple utterances in one turn. Some example dialogues are shown in Table 6 and Appendix I.

# dialogues	11157
# completed dialogues	9041
Vocabulary size	5325
Average # of utterances	11.41
Average time taken per task (sec.)	91.18
Average utterance length (tokens)	5.08
Number of linguistic templates[4]	41561

Table 3: Statistics of the MutualFriends dataset.

We categorize utterances into coarse types—inform, ask, answer, greeting, apology—by pattern matching (Appendix E). There are 7.4% multi-type utterances, and 30.9% utterances contain more than one entity. In Table 1, we show example utterances with rich semantics that cannot be sufficiently represented by traditional slot-values. Some of the standard ones are also non-trivial due to coreference and logical compositionality.

Our dataset also exhibits some interesting communication phenomena. Coreference occurs frequently when people check multiple attributes of one item. Sometimes mentions are dropped, as an utterance simply continues from the partner's utterance. People occasionally use external knowledge to group items with out-of-schema attributes (e.g., gender based on names, location based on schools). We summarize these phenomena in Table 2. In addition, we find 30% utterances involve cross-talk where the conversation does not progress linearly (e.g., italic utterances in Figure 1), a common characteristic of online chat (Ivanovic, 2005).

One strategic aspect of this task is choosing the order of attributes to mention. We find that people tend to start from attributes with fewer unique values, e.g., "all my friends like morning" given the KB_B in Table 6, as intuitively it would help exclude items quickly given fewer values to check.[5] We provide a more detailed analysis of strategy in Section 4.2 and Appendix F.

3 Dynamic Knowledge Graph Network

The diverse semantics in our data motivates us to combine unstructured representation of the dialogue history with structured knowledge. Our

[3] Tasks are put up in batches; the total time excludes intervals between batches.

[4] Entity names are replaced by their entity types.

[5] Our goal is to model human behavior thus we do not discuss the optimal strategy here.

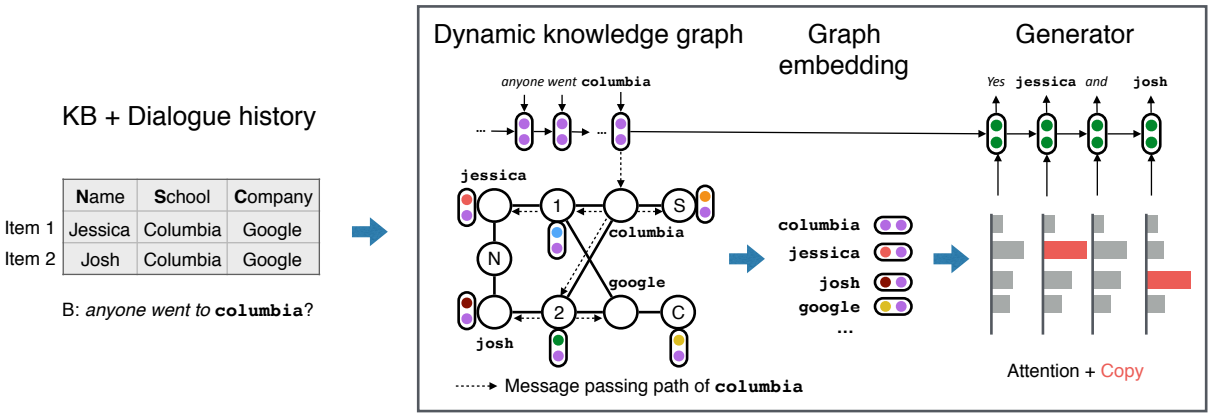

Figure 3: Overview of our approach. First, the KB and dialogue history (entities in **bold**) is mapped to a graph. Here, an item node is labeled by the item ID and an attribute node is labeled by the attribute's first letter. Next, each node is embedded using relevant utterance embeddings through message passing. Finally, an LSTM generates the next utterance based on attention over the node embeddings.

model consists of three components shown in Figure 3: (i) a dynamic knowledge graph, which represents the agent's private KB and shared dialogue history as a graph (Section 3.1), (ii) a graph embedding over the nodes (Section 3.2), and (iii) an utterance generator (Section 3.3).

The knowledge graph represents entities and relations in the agent's private KB, e.g., `item-1`'s `company` is `google`. As the conversation unfolds, utterances are embedded and incorporated into node embeddings of mentioned entities. For instance, in Figure 3, "anyone went to columbia" updates the embedding of `columbia`. Next, each node recursively passes its embedding to neighboring nodes so that related entities (e.g., those in the same row or column) also receive information from the most recent utterance. In our example, `jessica` and `josh` both receive new context when `columbia` is mentioned. Finally, the utterance generator, an LSTM, produces the next utterance by attending to the node embeddings.

3.1 Knowledge Graph

Given a dialogue of T utterances, we construct graphs $(G_t)_{t=1}^{T}$ over the KB and dialogue history for agent A.[6] There are three types of nodes: item nodes, attribute nodes, and entity nodes. Edges between nodes represent their relations. For example, `(item-1, hasSchool, columbia)` means that the first item has attribute `school` whose value is `columbia`. An example graph is shown in Figure 3. The graph G_t is updated based on utterance t by taking G_{t-1} and adding a new node for any entity mentioned in utterance t but not in KB_A.[7]

3.2 Graph Embedding

Given a knowledge graph, we are interested in computing a vector representation for each node v that captures both its unstructured context from the dialogue history and its structured context in the KB. A node embedding $V_t(v)$ for each node $v \in G_t$ is built from three parts: structural properties of an entity defined by the KB, embeddings of utterances in the dialogue history, and message passing between neighboring nodes.

Node Features. Simple structural properties of the KB often govern what is talked about; e.g., a high-frequency entity is usually interesting to mention (consider "All my friends like dancing."). We represent this type of information as a feature vector $F_t(v)$, which includes the degree and type (item, attribute, or entity type) of node v, and whether it has been mentioned in the current turn. Each feature is encoded as a one-hot vector and they are concatenated to form $F_t(v)$.

Mention Vectors. A mention vector $M_t(v)$ contains unstructured context from utterances relevant to node v up to turn t. To compute it, we first define the utterance representation $\tilde{u}_t$ and the set of relevant entities E_t. Let u_t be the embedding of utterance t (Section 3.3). To differentiate between

[6] It is important to differentiate perspectives of the two agents as they have different KBs. Thereafter we assume the perspective of agent A, i.e., accessing KB_A for A only, and refer to B as the partner.

[7] We use a rule-based lexicon to link text spans to entities. See details in Appendix D.

the agent's and the partner's utterances, we represent it as $\tilde{u}_t = \left[u_t \cdot \mathbb{1}_{\{u_t \in U_{\text{self}}\}}, u_t \cdot \mathbb{1}_{\{u_t \in U_{\text{partner}}\}}\right]$, where U_{self} and U_{partner} denote sets of utterances generated by the agent and the partner, and $[\cdot, \cdot]$ denotes concatenation. Let E_t be the set of entity nodes mentioned in utterance t if utterance t mentions some entities, or utterance $t-1$ otherwise.[8] The mention vector $M_t(v)$ of node v incorporates the current utterance if v is mentioned and inherits $M_{t-1}(v)$ if not:

$$M_t(v) = \lambda_t M_{t-1}(v) + (1-\lambda_t)\tilde{u}_t; \quad (1)$$

$$\lambda_t = \begin{cases} \sigma\left(W^{\text{inc}}\left[M_{t-1}(v), \tilde{u}_t\right]\right) & \text{if } v \in E_t, \\ 1 & \text{otherwise.} \end{cases}$$

Here, σ is the sigmoid function and W^{inc} is a parameter matrix.

Recursive Node Embeddings. We propagate information between nodes according to the structure of the knowledge graph. In Figure 3, given "anyone went to columbia?", the agent should focus on her friends who went to Columbia University. Therefore, we want this utterance to be sent to item nodes connected to `columbia`, and one step further to other attributes of these items because they might be mentioned next as relevant information, e.g., `jessica` and `josh`.

We compute the node embeddings recursively, analogous to belief propagation:

$$V_t^k(v) = \max_{v' \in N_t(v)} \tanh \left(W^{\text{mp}}\left[V_t^{k-1}(v'), R(e_{v \to v'})\right]\right), \quad (2)$$

where $V_t^k(v)$ is the depth-k node embedding at turn t and $N_t(v)$ denotes the set of nodes adjacent to v. The message from a neighboring node v' depends on its embedding at depth-$(k-1)$, the edge label $e_{v \to v'}$ (embedded by a relation embedding function R), and a parameter matrix W^{mp}. Messages from all neighbors are aggregated by $\max$, the element-wise max operation.[9] Example message passing paths are shown in Figure 3.

The final node embedding is the concatenation of embeddings at each depth:

$$V_t(v) = \left[V_t^0(v), \ldots, V_t^K(v)\right], \quad (3)$$

where K is a hyperparameter (we experiment with $K \in \{0, 1, 2\}$) and $V_t^0(v) = [F_t(v), M_t(v)]$.

[8] Relying on utterance $t-1$ is useful when utterance t answers a question, e.g., "do you have any google friends?" "No."

[9] Using sum or mean slightly hurts performance.

3.3 Utterance Embedding and Generation

We embed and generate utterances using Long Short Term Memory (LSTM) networks that take the graph embeddings into account.

Embedding. On turn t, upon receiving an utterance consisting of n_t tokens, $x_t = (x_{t,1}, \ldots, x_{t,n_t})$, the LSTM maps it to a vector as follows:

$$h_{t,j} = \text{LSTM}_{\text{enc}}(h_{t,j-1}, A_t(x_{t,j})), \quad (4)$$

where $h_{t,0} = h_{t-1,n_{t-1}}$, and A_t is an *entity abstraction* function, explained below. The final hidden state h_{t,n_t} is used as the utterance embedding u_t, which updates the mention vectors as described in Section 3.2.

In our dialogue task, the identity of an entity is unimportant. For example, replacing `google` with `alphabet` in Figure 1 should make little difference to the conversation. The role of an entity is determined instead by its relation to other entities and relevant utterances. Therefore, we define the abstraction $A_t(y)$ for a word y as follows: if y is linked to an entity v, then we represent an entity by its type (`school`, `company` etc.) embedding concatenated with its current node embedding: $A_t(y) = [E_{\text{type}(y)}, V_t(v)]$. Note that $V_t(v)$ is determined only by its structural features and its context. If y is a non-entity, then $A_t(y)$ is the word embedding of y concatenated with a zero vector of the same dimensionality as $V_t(v)$. This way, the representation of an entity only depends on its structural properties given by the KB and the dialogue context, which enables the model to generalize to unseen entities at test time.

Generation. Now, assuming we have embedded utterance x_{t-1} into $h_{t-1,n_{t-1}}$ as described above, we use another LSTM to generate utterance x_t. Formally, we carry over the last utterance embedding $h_{t,0} = h_{t-1,n_{t-1}}$ and define:

$$h_{t,j} = \text{LSTM}_{\text{dec}}(h_{t,j-1}, [A_t(x_{t,j}), c_{t,j}]), \quad (5)$$

where $c_{t,j}$ is a weighted sum of node embeddings in the current turn: $c_{t,j} = \sum_{v \in G_t} \alpha_{t,j,v} V_t(v)$, where $\alpha_{t,j,v}$ are the attention weights over the nodes. Intuitively, high weight should be given to relevant entity nodes as shown in Figure 3. We compute the weights through standard attention mechanism (Bahdanau et al., 2015):

$$\alpha_{t,j} = \text{softmax}(s_{t,j}),$$

$$s_{t,j,v} = w^{\text{attn}} \cdot \tanh\left(W^{\text{attn}}\left[h_{t,j-1}, V_t(v)\right]\right),$$

where vector w^{attn} and W^{attn} are parameters.

Finally, we define a distribution over both words in the vocabulary and nodes in G_t using the copying mechanism of Jia and Liang (2016):

$$p(x_{t,j+1} = y \mid G_t, x_{t,\leq j}) \propto \exp\left(W^{\text{vocab}} h_{t,j} + b\right),$$
$$p(x_{t,j+1} = r(v) \mid G_t, x_{t,\leq j}) \propto \exp\left(s_{t,j,v}\right),$$

where y is a word in the vocabulary, W^{vocab} and b are parameters, and $r(v)$ is the realization of the entity represented by node v, e.g., `google` is realized to "Google" during copying.[10]

4 Experiments

We compare our model with a rule-based system and a baseline neural model. Both automatic and human evaluations are conducted to test the models in terms of fluency, correctness, cooperation, and human-likeness. The results show that DynoNet is able to converse with humans in a coherent and strategic way.

4.1 Setup

We randomly split the data into train, dev, and test sets (8:1:1). We use a one-layer LSTM with 100 hidden units and 100-dimensional word vectors for both the encoder and the decoder (Section 3.3). Each successful dialogue is turned into two examples, each from the perspective of one of the two agents. We maximize the log-likelihood of all utterances in the dialogues. The parameters are optimized by AdaGrad (Duchi et al., 2010) with an initial learning rate of 0.5. We trained for at least 10 epochs; after that, training stops if there is no improvement on the dev set for 5 epochs. By default, we perform $K = 2$ iterations of message passing to compute node embeddings (Section 3.2). For decoding, we sequentially sample from the output distribution with a softmax temperature of 0.5.[11] Hyperparameters are tuned on the dev set.

We compare DynoNet with its static cousion (StanoNet) and a rule-based system (Rule). StanoNet uses G_0 throughout the dialogue, thus the dialogue history is completely contained in the LSTM states instead of being injected into the knowledge graph. Rule maintains weights for each entity and each item in the KB to decide what to talk about and which item to select. It has a pattern-matching semantic parser, a rule-based policy, and a templated generator. See Appendix G for details.

4.2 Evaluation

We test our systems in two interactive settings: bot-bot chat and bot-human chat. We perform both automatic evaluation and human evaluation.

Automatic Evaluation. First, we compute the cross-entropy (ℓ) of a model on test data. As shown in Table 4, DynoNet has the lowest test loss. Next, we have a model chat with itself on the scenarios from the test set.[12] We evaluate the chats with respect to language variation, effectiveness, and strategy.

For language variation, we report the average utterance length L_u and the unigram entropy H in Table 4. Compared to Rule, the neural models tend to generate shorter utterances (Li et al., 2016b; Serban et al., 2017b). However, they are more diverse; for example, questions are asked in multiple ways such as "Do you have ...", "Any friends like ...", "What about ...".

At the discourse level, we expect the distribution of a bot's utterance types to match the distribution of human's. We show percentages of each utterance type in Table 4. For Rule, the decision about which action to take is written in the rules, while StanoNet and DynoNet learned to behave in a more human-like way, frequently informing and asking questions.

To measure effectiveness, we compute the overall success rate (C) and the success rate per turn (C_T) and per selection (C_S). As shown in Table 4, humans are the best at this game, followed by Rule which is comparable to DynoNet.

Next, we investigate the strategies leading to these results. An agent needs to decide which entity/attribute to check first to quickly reduce the search space. We hypothesize that humans tend to first focus on a majority entity and an attribute with fewer unique values (Section 2.3). For example, in the scenario in Table 6, `time` and `location` are likely to be mentioned first. We show the average frequency of first-mentioned entities (#Ent$_1$) and the average number of unique values for first-mentioned attributes ($|\text{Attr}_1|$) in Ta-

[10] We realize an entity by sampling from the empirical distribution of its surface forms found in the training data.

[11] Since selection is a common 'utterance' in our dataset and neural generation models are susceptible to over-generating common sentences, we halve its probability during sampling.

[12] We limit the number of turns in bot-bot chat to be the maximum number of turns humans took in the test set (46 turns).

System	$\ell \downarrow$	L_u	H	$C \uparrow$	$C_T \uparrow$	$C_S \uparrow$	Sel	Inf	Ask	Ans	Greet	#Ent$_1$	|Attr$_1$|	#Ent	#Attr
Human	-	5.10	4.57	.82	.07	.38	.21	.31	.17	.08	.08	.55	.35	6.1	2.6
Rule	-	7.61	3.37	.90	.05	**.29**	.18	**.34**	.23	.00	.12	.24	.61	9.9	3.0
StanoNet	2.20	4.01	**4.05**	.78	.04	.18	.19	.26	.12	.23	**.09**	.61	**.19**	7.1	2.9
DynoNet	**2.13**	3.37	3.90	**.96**	**.06**	.25	**.22**	.26	**.13**	.20	.12	**.55**	.18	**5.2**	**2.5**

Table 4: Automatic evaluation on human-human and bot-bot chats on test scenarios. We use ↑ / ↓ to indicate that higher / lower values are better; otherwise the objective is to match humans' statistics. Best results (except Human) are in bold. Neural models generate shorter (lower L_u) but more diverse (higher H) utterances. Overall, their distributions of utterance types match those of the humans'. (We only show the most frequent speech acts therefore the numbers do not sum to 1.) Rule is effective in completing the task (higher C_S), but it is not information-efficient given the large number of attributes (#Attr) and entities (#Ent) mentioned.

ble 4.[13] Both DynoNet and StanoNet successfully match human's starting strategy by favoring entities of higher frequency and attributes of smaller domain size.

To examine the overall strategy, we show the average number of attributes (#Attr) and entities (#Ent) mentioned during the conversation in Table 4. Humans and DynoNet strategically focus on a few attributes and entities, whereas Rule needs almost twice entities to achieve similar success rates. This suggests that the effectiveness of Rule mainly comes from large amounts of unselective information, which is consistent with comments from their human partners.

Partner Evaluation. We generated 200 new scenarios and put up the bots on AMT using the same chat interface that was used for data collection. The bots follow simple turn-taking rules explained in Appendix H. Each AMT worker is randomly paired with Rule, StanoNet, DynoNet, or another human (but the worker doesn't know which), and we make sure that all four types of agents are tested in each scenario at least once. At the end of each dialogue, humans are asked to rate their partner in terms of fluency, correctness, cooperation, and human-likeness from 1 (very bad) to 5 (very good), along with optional comments.

We show the average ratings (with significance tests) in Table 5 and the histograms in Appendix J. In terms of fluency, the models have similar performance since the utterances are usually short. Judgment on correctness is a mere guess since the evaluator cannot see the partner's KB; we will analyze correctness more meaningfully in the third-party evaluation below.

[13] Both numbers are normalized to $[0, 1]$ with respect to all entities/attributes in the corresponding KB.

Noticeably, DynoNet is more cooperative than the other models. As shown in the example dialogues in Table 6, DynoNet cooperates smoothly with the human partner, e.g., replying with relevant information about morning/indoor friends when the partner mentioned that all her friends prefer morning and most like indoor. StanoNet starts well but doesn't follow up on the morning friend, presumably because the `morning` node is not updated dynamically when mentioned by the partner. Rule follows the partner poorly. In the comments, the biggest complaint about Rule was that it was not 'listening' or 'understanding'. Overall, DynoNet achieves better partner satisfaction, especially in cooperation.

Third-party Evaluation. We also created a *third-party evaluation* task, where an independent AMT worker is shown a conversation and the KB of one of the agents; she is asked to rate the same aspects of the agent as in the partner evaluation and provide justifications. Each agent in a dialogue is rated by at least 5 people.

The average ratings and histograms are shown in Table 5 and Appendix J. For correctness, we see that Rule has the best performance since it always tells the truth, whereas humans can make mistakes due to carelessness and the neural models can generate false information. For example, in Table 6, DynoNet 'lied' when saying that it has a morning friend who likes outdoor.

Surprisingly, there is a discrepancy between the two evaluation modes in terms of cooperation and human-likeness. Manual analysis of the comments indicates that third-party evaluators focus less on the dialogue strategy and more on linguistic features, probably because they were not fully engaged in the dialogue. For example, justification

System	C	C_T	C_S	Partner eval				Third-party eval			
				Flnt	Crct	Coop	Human	Flnt	Crct	Coop	Human
Human	.89	.07	.36	4.2^{rds}	4.3^{rds}	4.2^{rds}	4.1^{rds}	4.0	4.3^{ds}	4.0^{ds}	4.1^{rds}
Rule	**.88**	**.06**	**.29**	3.6	4.0	3.5	3.5	4.0	$\mathbf{4.4}^{hds}$	$\mathbf{3.9}^{s}$	$\mathbf{4.0}^{s}$
StanoNet	.76	.04	.23	3.5	3.8	3.4	3.3	4.0	4.0	3.8	3.8
DynoNet	.87	.05	.27	$\mathbf{3.8}^{s}$	4.0	$\mathbf{3.8}^{rs}$	$\mathbf{3.6}^{s}$	4.0	4.1	3.9	3.9

Table 5: Results on human-bot/human chats. Best results (except Human) in each column are in bold. We report the average ratings of each system. For third-party evaluation, we first take mean of each question then average the ratings. DynoNet has the best partner satisfaction in terms of fluency (Flnt), correctness (Crct), cooperation (Coop), human likeness (Human). The superscript of a result indicates that its advantage over other systems (r: Rule, s: StanoNet, d: DynoNet) is statistically significant with $p < 0.05$ given by paired t-tests.

for cooperation often mentions frequent questions and timely answers, less attention is paid to what is asked about though.

For human-likeness, partner evaluation is largely correlated with coherence (e.g., not repeating or ignoring past information) and task success, whereas third-party evaluators often rely on informality (e.g., usage of colloquia like "hiya", capitalization, and abbreviation) or intuition. Interestingly, third-party evaluators noted most phenomena listed in Table 2 as indicators of human-beings, e.g., correcting oneself, making chit-chat other than simply finishing the task. See example comments in Appendix K.

4.3 Ablation Studies

Our model has two novel designs: entity abstraction and message passing for node embeddings. Table 7 shows what happens if we ablate these. When the number of message passing iterations, K, is reduced from 2 to 0, the loss consistently increases. Removing entity abstraction—meaning adding entity embeddings to node embeddings and the LSTM input embeddings—also degrades performance. This shows that DynoNet benefits from contextually-defined, structural node embeddings rather than ones based on a classic lookup table.

Model	ℓ
DynoNet (K = 2)	**2.16**
DynoNet (K = 1)	2.20
DynoNet (K = 0)	2.26
DynoNet (K = 2) w/o entity abstraction	2.21

Table 7: Ablations of our model on the dev set show the importance of entity abstraction and message passing ($K = 2$).

5 Discussion and Related Work

There has been a recent surge of interest in end-to-end task-oriented dialogue systems, though progress has been limited by the size of available datasets (Serban et al., 2015a). Most work focuses on information-querying tasks, using Wizard-of-Oz data collection (Williams et al., 2016; Asri et al., 2016) or simulators (Bordes and Weston, 2017; Li et al., 2016d), In contrast, collaborative dialogues are easy to collect as natural human conversations, and are also challenging enough given the large number of scenarios and diverse conversation phenomena. There are some interesting strategic dialogue datasets—settlers of Catan (Afantenos et al., 2012) (2K turns) and the cards corpus (Potts, 2012) (1.3K dialogues), as well as work on dialogue strategies (Keizer et al., 2017; Vogel et al., 2013), though no full dialogue system has been built for these datasets.

Most task-oriented dialogue systems follow the POMDP-based approach (Williams and Young, 2007; Young et al., 2013). Despite their success (Wen et al., 2017; Dhingra et al., 2017; Su et al., 2016), the requirement for handcrafted slots limits their scalability to new domains and burdens data collection with extra state labeling. To go past this limit, Bordes and Weston (2017) proposed a Memory-Networks-based approach without domain-specific features. However, the memory is unstructured and interfacing with KBs relies on API calls, whereas our model embeds both the dialogue history and the KB structurally. Williams et al. (2017) use an LSTM to automatically infer the dialogue state, but as they focus on dialogue control rather than the full problem, the response is modeled as a templated action, which restricts the generation of richer utterances. Our network ar-

Friends of A

ID	Name	Company	Time	Location
1	Kathy	TRT Holdings	afternoon	indoor
2	Jason	Dollar General	afternoon	indoor
3	Johnny	TRT Holdings	afternoon	outdoor
4	Frank	SFN Group	afternoon	indoor
5	Catherine	Dollar General	afternoon	indoor
6	Catherine	Weis Markets	afternoon	indoor
7	Kathleen	TRT Holdings	morning	indoor
8	Lori	TRT Holdings	afternoon	indoor
9	Frank	L&L Hawaiian Barbecue	afternoon	outdoor

Friends of B

ID	Name	Company	Time	Location
1	Justin	New Era Tickets	morning	indoor
2	Kathleen	TRT Holdings	morning	indoor
3	Gloria	L&L Hawaiian Barbecue	morning	indoor
4	Kathleen	Advance Auto Parts	morning	outdoor
5	Justin	Arctic Cat	morning	indoor
6	Anna	Dollar General	morning	indoor
7	Steven	SFN Group	morning	indoor
8	Wayne	R.J. Corman Railroad Group	morning	indoor
9	Alexander	R.J. Corman Railroad Group	morning	indoor

A: Human B: Human

A: Hi
B: hey || i have one outdoor
A: I have 4 TRT Holdings || I have 2 outdoor one Johnny, other Frank
B: i only have one TRT Holdings - Kathleen
A: SELECT 7 B: SELECT 2

A: DynoNet B: Human

A: **hi**
B: Hello || all my friends prefer morning
A: **1 of my morning likes the outdoors**
B: and all like indoor except for one
A: **do they work for trt holdings?**
B: Kathleen?
A: **SELECT 7** B: SELECT 2

A: StanoNet B: Human

A: **Hello**
B: hi
A: **Hello || I have one morning person.**
B: all of my friends like mornings
A: **My friend prefers afternoon works at trt holdings.**
B: what is their name?
A: **Likes indoors.**
B: what is your fiend who likes morning name?
A: **They work for trt holdings.**
B: SELECT 2 A: **SELECT 7**

A: Human B: Rule

B: **hiya**
A: hEY
B: **I have 1 indoors and kathleen.**
A: Most of mine are indoors.
B: **SELECT 1**
A: I have one morning and rest afternoon.
B: **Do you have any friend working at l hawaiian?**
A: I don't know Justin
B: **I have 1 alexander.**
...

Table 6: Examples of human-bot chats. The mutual friend is highlighted in blue in each KB. Bots' utterances are in bold and selected items are represented by item IDs. Only the first half of the human-Rule chat is shown due to limited space. Multiple utterances of one agent rae separated by ||.

chitecture is most similar to EntNet (Henaff et al., 2017), where memories are also updated by input sentences recurrently. The main difference is that our model allows information to be propagated between structured entities, which is shown to be crucial in our setting (Section 4.3).

Our work is also related to language generation conditioned on knowledge bases (Mei et al., 2016; Kiddon et al., 2016). One challenge here is to avoid generating false or contradicting statements, which is currently a weakness of neural models. Our model is mostly accurate when generating facts and answering existence questions about a single entity, but will need a more advanced attention mechanism for generating utterances involving multiple entities, e.g., attending to items or attributes first, then selecting entities; generating high-level concepts before composing them to natural tokens (Serban et al., 2017a).

In conclusion, we believe the symmetric collaborative dialogue setting and our dataset provide unique opportunities at the interface of traditional task-oriented dialogue and open-domain chat. We also offered DynoNet as a promising means for open-ended dialogue state representation. Our dataset facilitates the study of pragmatics and human strategies in dialogue—a good stepping stone towards learning more complex dialogues such as negotiation.

Acknowledgments. This work is supported by DARPA Communicating with Computers (CwC) program under ARO prime contract no. W911NF-15-1-0462. Mike Kayser worked on an early version of the project while he was at Stanford. We also thank members of the Stanford NLP group for insightful discussions.

Reproducibility. All code, data, and experiments for this paper are available on the CodaLab platform: https://worksheets.codalab.org/worksheets/0xc757f29f5c794e5eb7bfa8ca9c945573.

References

S. Afantenos, N. Asher, F. Benamara, A. Cadilhac, C. Dégremont, P. Denis, M. Guhe, S. Keizer, A. Lascarides, O. Lemon, P. Muller, S. Paul, V. Rieser, and L. Vieu. 2012. Developing a corpus of strategic conversation in the settlers of catan. In *SeineDial 2012 - The 16th Workshop on the Semantics and Pragmatics of Dialogue*.

L. E. Asri, H. Schulz, S. Sharma, J. Zumer, J. Harris, E. Fine, R. Mehrotra, and K. Suleman. 2016. Frames: A corpus for adding memory to goal-oriented dialogue systems. *Maluuba Technical Report* .

D. Bahdanau, K. Cho, and Y. Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*.

A. Bordes and J. Weston. 2017. Learning end-to-end goal-oriented dialog. In *International Conference on Learning Representations (ICLR)*.

B. Dhingra, L. Li, X. Li, J. Gao, Y. Chen, F. Ahmed, and L. Deng. 2017. End-to-end reinforcement learning of dialogue agents for information access. In *Association for Computational Linguistics (ACL)*.

J. Duchi, E. Hazan, and Y. Singer. 2010. Adaptive subgradient methods for online learning and stochastic optimization. In *Conference on Learning Theory (COLT)*.

M. Henaff, J. Weston, A. Szlam, A. Bordes, and Y. LeCun. 2017. Tracking the world state with recurrent entity networks. In *International Conference on Learning Representations (ICLR)*.

E. Ivanovic. 2005. Dialogue act tagging for instant messaging chat sessions. In *Association for Computational Linguistics (ACL)*.

R. Jia and P. Liang. 2016. Data recombination for neural semantic parsing. In *Association for Computational Linguistics (ACL)*.

S. Keizer, M. Guhe, H. Cuayahuitl, I. Efstathiou, K. Engelbrecht, M. Dobre, A. Lascarides, and O. Lemon. 2017. Evaluating persuasion strategies and deep reinforcement learning methods for negotiation dialogue agents. In *European Association for Computational Linguistics (EACL)*.

C. Kiddon, L. S. Zettlemoyer, and Y. Choi. 2016. Globally coherent text generation with neural checklist models. In *Empirical Methods in Natural Language Processing (EMNLP)*.

J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. 2016a. A persona-based neural conversation model. In *Association for Computational Linguistics (ACL)*.

J. Li, M. Galley, C. Brockett, J. Gao, and W. B. Dolan. 2016b. A diversity-promoting objective function for neural conversation models. In *Human Language Technology and North American Association for Computational Linguistics (HLT/NAACL)*.

J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao. 2016c. Deep reinforcement learning for dialogue generation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

X. Li, Z. C. Lipton, B. Dhingra, L. Li, J. Gao, and Y. Chen. 2016d. A user simulator for task-completion dialogues. *arXiv* .

C. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

R. T. Lowe, N. Pow, I. Serban, L. Charlin, C. Liu, and J. Pineau. 2017. Training end-to-end dialogue systems with the ubuntu dialogue corpus. *Dialogue and Discourse* 8.

H. Mei, M. Bansal, and M. R. Walter. 2016. What to talk about and how? selective generation using LSTMs with coarse-to-fine alignment. In *Human Language Technology and North American Association for Computational Linguistics (HLT/NAACL)*.

H. Mei, M. Bansal, and M. R. Walter. 2017. Coherent dialogue with attention-based language models. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

C. Potts. 2012. Goal-driven answers in the Cards dialogue corpus. In *Proceedings of the 30th West Coast Conference on Formal Linguistics*.

I. Serban, T. Klinger, G. Tesauro, K. Talamadupula, B. Zhou, Y. Bengio, and A. C. Courville. 2017a. Multiresolution recurrent neural networks: An application to dialogue response generation. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

I. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. C. Courville, and Y. Bengio. 2017b. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

I. V. Serban, R. Lowe, L. Charlin, and J. Pineau. 2015a. A survey of available corpora for building data-driven dialogue systems. *arXiv preprint arXiv:1512.05742* .

I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. 2015b. Building end-to-end dialogue systems using generative hierarchical neural network models. *arXiv preprint arXiv:1507.04808* .

L. Shang, Z. Lu, and H. Li. 2015. Neural responding machine for short-text conversation. In *Association for Computational Linguistics (ACL)*.

A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J. Nie, J. Gao, and B. Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *North American Association for Computational Linguistics (NAACL)*.

P. Su, M. Gasic, N. Mrksic, L. M. Rojas-Barahona, S. Ultes, D. Vandyke, T. Wen, and S. J. Young. 2016. Continuously learning neural dialogue management. *arXiv preprint arXiv:1606.02689* .

A. Vogel, M. Bodoia, C. Potts, and D. Jurafsky. 2013. Emergence of gricean maxims from multi-agent decision theory. In *North American Association for Computational Linguistics (NAACL)*. pages 1072–1081.

T. Wen, M. Gasic, N. Mrksic, L. M. Rojas-Barahona, P. Su, S. Ultes, D. Vandyke, and S. Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *European Association for Computational Linguistics (EACL)*.

J. D. Williams, K. Asadi, and G. Zweig. 2017. Hybrid code networks: Practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Association for Computational Linguistics (ACL)*.

J. D. Williams, A. Raux, and M. Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue and Discourse* 7.

J. D. Williams and S. Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language* 21(2):393–422.

S. Young, M. Gasic, B. Thomson, and J. D. Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE* 101(5):1160–1179.

A Knowledge Base Schema

The attribute set $\mathcal{A}$ for the MutualFriends task contains name, school, major, company, hobby, time-of-day preference, and location preference. Each attribute a has a set of possible values (entities) $\mathcal{E}_a$. For name, school, major, company, and hobby, we collected a large set of values from various online sources.[14] We used three possible values (morning, afternoon, and evening) for the time-of-day preference, and two possible values (indoors and outdoors) for the location preference.

B Scenario Generation

We generate scenarios randomly to vary task complexity and elicit linguistic and strategic variants. A scenario S is characterized by the number of items (N_S), the attribute set ($\mathcal{A}_S$) whose size is M_S, and the values for each attribute $a \in \mathcal{A}_S$ in the two KBs.

A scenario is generated as follows.

1. Sample N_S and M_S uniformly from $\{5, \ldots, 12\}$ and $\{3, 4\}$ respectively.
2. Generate $\mathcal{A}_S$ by sampling M_S attributes without replacement from $\mathcal{A}$.
3. For each attribute $a \in \mathcal{A}_S$, sample the concentration parameter α_a uniformly from the set $\{0.3, 1, 3\}$.
4. Generate two KBs by sampling N_S values for each attribute a from a Dirichlet-multinomial distribution over the value set $\mathcal{E}_a$ with the concentration parameter α_a.

We repeat the last step until the two KBs have one unique common item.

C Chat Interface

In order to collect real-time dialogue between humans, we set up a web server and redirect AMT workers to our website. Visitors are randomly paired up as they arrive. For each pair, we choose a random scenario, and randomly assign a KB to each dialogue participant. We instruct people to play intelligently, to refrain from brute-force tactics (e.g., mentioning every attribute value), and to use grammatical sentences. To discourage random guessing, we prevent users from selecting a friend (item) more than once every 10 seconds. Each worker was paid $0.35 for a successful dialogue within a 5-minute time limit. We log each utterance in the dialogue along with timing information.

D Entity Linking and Realization

We use a rule-based lexicon to link text spans to entities. For every entity in the schema, we compute different variations of its canonical name, including acronyms, strings with a certain edit distance, prefixes, and morphological variants. Given a text span, a set of candidate entities is returned by string matching. A heuristic ranker then scores each candidate (e.g., considering whether the span is a substring of a candidate, the edit distance between the span and a candidate etc.). The highest-scoring candidate is returned.

A linked entity is considered as a single token and its surface form is ignored in all models. At generation time, we realize an entity by sampling from the empirical distribution of its surface forms in the training set.

E Utterance Categorization

We categorize utterances into inform, ask, answer, greeting, apology heuristically by pattern matching.

- An ask utterance asks for information regarding the partner's KB. We detect these utterances by checking for the presence of a '?' and/or a question word like "do", "does", "what", etc.
- An inform utterance provides information about the agent's KB. We define it as an utterances that mentions entities in the KB and is not an ask utterance.
- An answer utterance simply provides a positive/negative response to a question, containing words like "yes", "no", "nope", etc.
- A greeting utterance contains words like "hi" or "hello"; it often occurs at the beginning of a dialogue.

[14] Names: https://www.ssa.gov/oact/babynames/decades/century.html
Schools: http://doors.stanford.edu/~sr/universities.html
Majors: http://www.a2zcolleges.com/majors
Companies: https://en.wikipedia.org/wiki/List_of_companies_of_the_United_States
Hobbies: https://en.wikipedia.org/wiki/List_of_hobbies

- An apology utterance contains the word "sorry", which is typically associated with corrections and wrong selections.

See Table 2 and Table 1 for examples of these utterance types.

F Strategy

During scenario generation, we varied the number of attributes, the number of items in each KB, and the distribution of values for each attribute. We find that as the number of items and/or attributes grows, the dialogue length and the completion time also increase, indicating that the task becomes harder. We also anticipated that varying the value of α would impact the overall strategy (for example, the order in which attributes are mentioned) since α controls the skewness of the distribution of values for an attribute.

On examining the data, we find that humans tend to first mention attributes with a more skewed (i.e., less uniform) distribution of values. Specifically, we rank the α values of all attributes in a scenario (see step 3 in Section B), and bin them into 3 distribution groups—least_uniform, medium, and most_uniform, according to the ranking where higher α values corresponds to more uniform distributions.[15] In Figure 4, we plot the histogram of the distribution group of the first-mentioned attribute in a dialogues, which shows that skewed attributes are mentioned much more frequently.

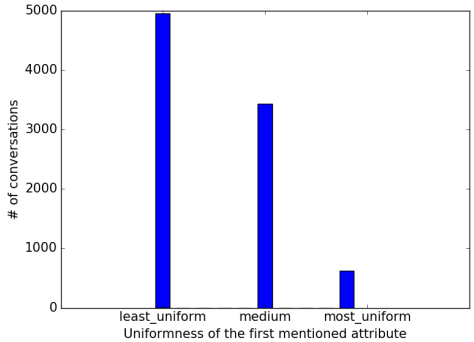

Figure 4: Histogram of the first attribute mentioned in a dialogue. People tend to first mention attributes from very skewed (non-uniform) distributions.

[15] For scenarios with 3 attributes, each group contains one attributes. For scenarios with 4 attributes, we put the two attributes with rankings in the middle to medium.

G Rule-based System

The rule-based bot takes the following actions: greeting, informing or asking about a set of entities, answering a question, and selecting an item. The set of entities to inform/ask is sampled randomly given the entity weights. Initially, each entity is weighted by its count in the KB. We then increment or decrement weights of entities mentioned by the partner and its related entities (in the same row or column), depending on whether the mention is positive or negative. A negative mention contains words like "no", "none", "n't" etc. Similarly, each item has an initial weight of 1, which is updated depending on the partner's mention of its attributes.

If there exists an item with weight larger than 1, the bot selects the highest-weighted item with probability 0.3. If a question is received, the bot informs facts of the entities being asked, e.g., "anyone went to columbia?", "I have 2 friends who went to columbia". Otherwise, the bot samples an entity set and randomly chooses between informing and asking about the entities.

All utterances are generated by sentence templates, and parsing of the partner's utterance is done by entity linking and pattern matching (Section E).

H Turn-taking Rules

Turn-taking is universal in human conversations and the bot needs to decide when to 'talk' (send an utterance). To prevent the bot from generating utterances continuously and forming a monologue, we allow it to send at most one utterance if the utterance contains any entity, and two utterances otherwise. When sending more than one utterance in a turn, the bot must wait for 1 to 2 seconds in between. In addition, after an utterance is generated by the model (almost instantly), the bot must hold on for some time to simulate message typing before sending. We used a typing speed of 7 chars / sec and added an additional random delay between 0 to 1.5s after 'typing'. The rules are applied to all models.

I Additional Human-Bot Dialogue

We show another set of human-bot/human chats in Table 8. In this scenario, the distribution of values are more uniform compared to Table 6. Nevertheless, we see that StanoNet and DynoNet

still learned to start from relatively high-frequency entities. They also appear more cooperative and mentions relevant entities in the dialogue context compared to Rule.

J Histograms of Ratings from Human Evaluations

The histograms of ratings from partner and third-party evaluations is shown in Figure 5 and Figure 6 respectively. As these figures show, there are some obvious discrepancies between the ratings made by agents who chatted with the bot and those made by an 'objective' third party. These ratings provide some interesting insights into how dialogue participants in this task setting perceive their partners, and what constitutes a 'human-like' or a 'fluent' partner.

K Example Comments from Partner and Third-party Evaluations

In Table 9, we show several pairs of ratings and comments on human-likeness for the same dialogue from both the partner evaluation and the third-party evaluation. As a conversation participant, the dialogue partner often judges from the cooperation and strategy perspective, whereas the third-party evaluator relies more on linguistic features (e.g., length, spelling, formality).

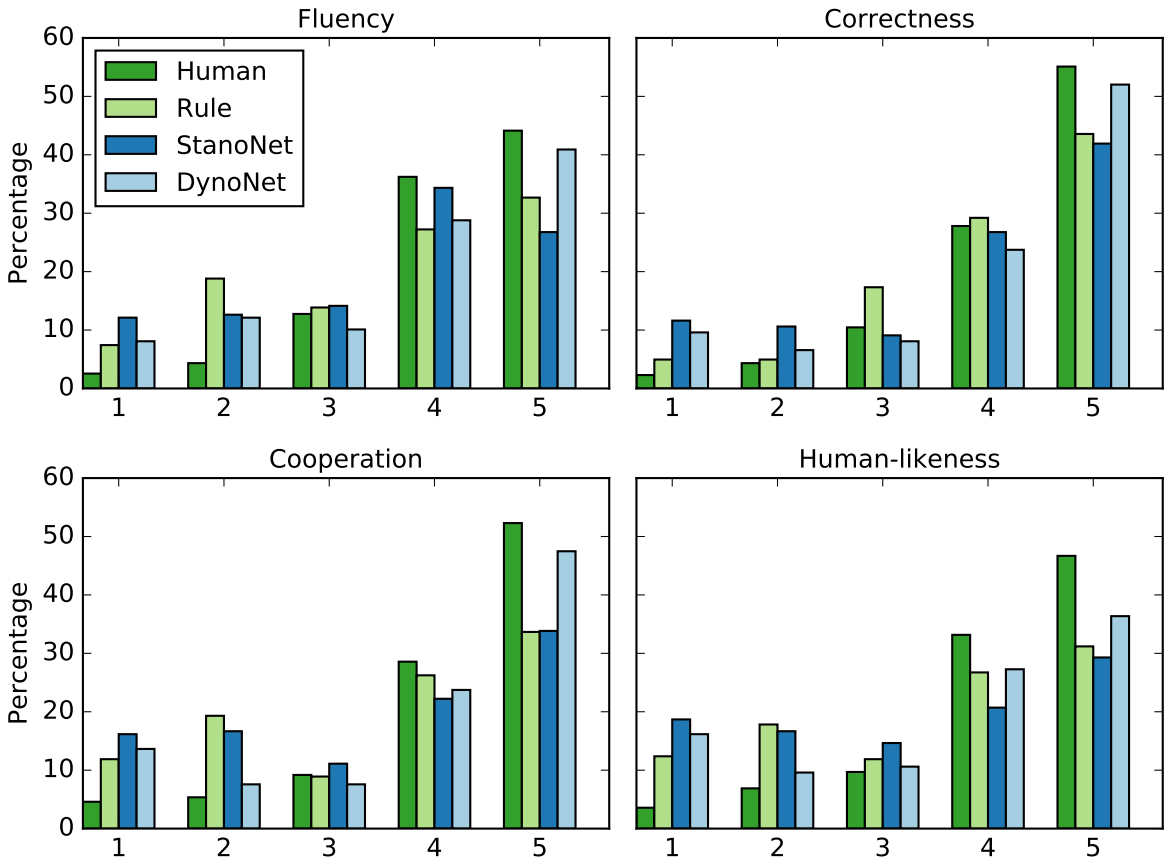

Figure 5: Histogram of ratings (higher is better) from dialogue partners. DynoNet is better than all other systems, especially in cooperation.

Friends of A

ID	Major	Company	Hobby
1	Metallurgical Engineering	Gannett Company	Candle making
2	Business Education	Electronic Arts	Gunsmithing
3	Parks Administration	Kenworth	Water sports
4	Mathematics Education	Electronic Arts	Astronomy
5	Agricultural Mechanization	AVST	Field hockey
6	Mathematics Education	AVST	Shopping
7	Parks Administration	Adobe Systems	Foreign language learning
8	Agricultural Mechanization	Bronco Wine Company	Shopping
9	Metallurgical Engineering	Electronic Arts	Foreign language learning
10	Mathematics Education	Electronic Arts	Poi

Friends of B

ID	Major	Company	Hobby
1	Foreign Language Teacher Education	Gannett Company	Road biking
2	Mathematics Education	Electronic Arts	Astronomy
3	Petroleum Engineering	Western Sugar Cooperative	Candle making
4	Mathematics Education	American Broadcasting Company	Road biking
5	Petroleum Engineering	Western Sugar Cooperative	Road biking
6	Petroleum Engineering	A& W Restaurants	Golfing
7	Petroleum Engineering	American Broadcasting Company	Origami
8	Russian	The Walt Disney Company	Astronomy
9	Petroleum Engineering	The Walt Disney Company	Origami
10	Protestant Affiliation	Acme Brick	Astronomy

A: Human B: Human	A: Human B: DynoNet
A: hi B: hi A: Any friends work at AVST? B: petroleum engi?? || no A: no petroleum B: russian as major? A: Nope B: protestant afiil? || math? A: I have two math B: SELECT 2 A: one works at Electronic Arts || SELECT 10 B: SELECT 1 A: SELECT 4 B: SELECT 2	B: **hi** A: hey B: **i have three friends who like road biking** A: i have two friends who like foreign language learning B: **no foreign language learning here** A: I have no friends who like road biking B: **do they major in foreign language?** A: No || no B: **petroleum engineering**? A: No but I have four friends who work for the Electronic Arts company B: **SELECT 2** A: SELECT 4
A: StanoNet B: Human	**A: Human B: Rule**
A: **hi** || **do you have any friends that work at electronic arts?** B: One and they like astronomy A: **SELECT 10** B: SELECT 2 A: **SELECT 4**	B: **hiya** || **i have one foreign language and gannett** A: i have two avst B: **do you have any acme brick and astronomy?** A: many maths people || two are foreign language B: **do you have any petroleum engineering and american broadcasting company?** A: no || electronic arts B: **SELECT 1** A: avst B: **do you have any disney or restaurant?** ...

Table 8: Example human-bot chats. The mutual friend is highlighted in blue in each KB. Bots' utterances are in bold and selected items are represented by item IDs. Only the first half of the human-Rule chat is shown due to space limit. Multiple utterances of one agent is separated by ||.

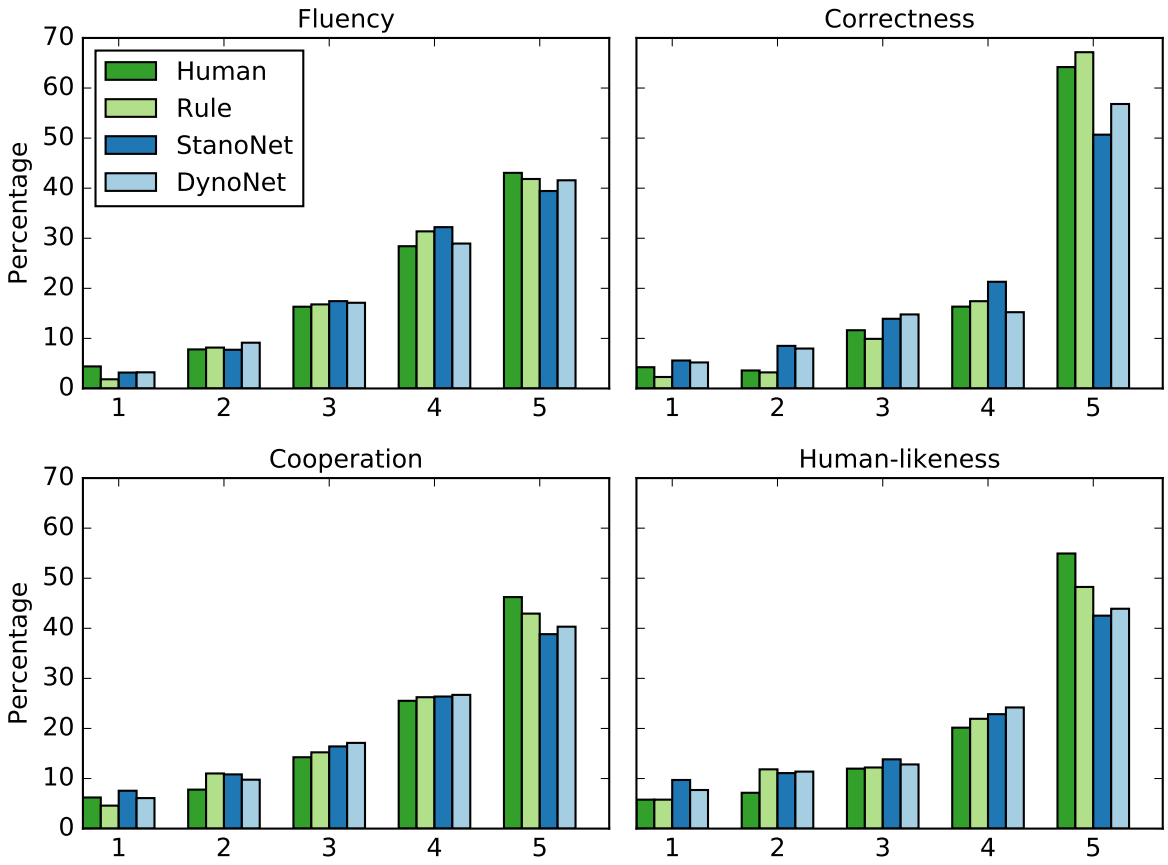

Figure 6: Histogram of ratings (higher is better) from third-party evaluators. Differences between systems are less significant.

System	Partner evaluation (1 per dialogue)		Third-party evaluation (5 per dialogue)	
	Human	Comments	Human	Justifications
Human	4	Good partner. Easy to work with	4.6	- you have any friends who went to monmouth? - The flow was nice and they were able to discern the correct answers. - human like because of interaction talking - Answers are human like, not robotic. Uses ”hiya” to begin conversation, more of a warm tone. - more human than computer Agent 2: hiya Agent 1: Hey
Rule	2	Didn’t listen to me	4	- agent 2 looked human to me - definitely human - A2 could be replaced with a robot without noticeable difference. - They spoke and behaved as I or any human would in this situation. - The agent just seems to be going through the motions, which gives me the idea that the agent doesn’t exbit humanlike characteristics.
StanoNet	5	Took forever and didn’t really respond correctly to questions.	3.5	- No djarum – This doesn’t make sense in this context, so doesn’t seem to be written by a human. - human like because of slight mispellingss - Can tell they are likely human but just not very verbose - Their terse conversion leans to thinking they were either not paying attention or not human. - The short vague sentences are very human like mistakes.
DynoNet	4	I replied twice that I only had indoor friends and was ignored.	3.8	- Agent 1 is very human like based on the way they typed and the fact that they were being deceiving. - Pretty responsive and logical progression, but it’s very stilted sounding - i donot have a jose - Agent gives normal human responses, “no angela i don’t” - agent 1 was looking like a humanlike

Table 9: Comparison of ratings and comments on human-likeness from partners and third-party evaluators. Each row contains results for the same dialogue. For the partner evaluation, we ask the human partner to provide a single, optional comment at the end of the conversation. For the third-party evaluation, we ask five Turkers to rate each dialogue and report the mean score; they must provide justification for ratings in each aspect. From the comments, we see that dialogue partners focus more on cooperation and effectiveness, whereas third-party evaluators focus more on linguistic features such as verbosity and informality.