Deal or No Deal? End-to-End Learning for Negotiation Dialogues

Mike Lewis\textsuperscript{1}, Denis Yarats\textsuperscript{1}, Yann N. Dauphin\textsuperscript{1}, Devi Parikh\textsuperscript{2,1} and Dhruv Batra\textsuperscript{2,1}
\textsuperscript{1}Facebook AI Research \textsuperscript{2}Georgia Institute of Technology
\{mikelewis,denisy,ynd\}@fb.com \{parikh,dbatra\}@gatech.edu

Abstract
Much of human dialogue occurs in semi-cooperative settings, where agents with different goals attempt to agree on common decisions. Negotiations require complex communication and reasoning skills, but success is easy to measure, making this an interesting task for AI. We gather a large dataset of human-human negotiations on a multi-issue bargaining task, where agents who cannot observe each other’s reward functions must reach an agreement (or a deal) via natural language dialogue. For the first time, we show it is possible to train end-to-end models for negotiation, which must learn both linguistic and reasoning skills with no annotated dialogue states. We also introduce dialogue rollouts, in which the model plans ahead by simulating possible complete continuations of the conversation, and find that this technique dramatically improves performance. Our code and dataset are publicly available\footnote{https://github.com/facebookresearch/end-to-end-negotiator}.

1 Introduction
Intelligent agents often need to cooperate with others who have different goals, and typically use natural language to agree on decisions. Negotiation is simultaneously a linguistic and a reasoning problem, in which an intent must be formulated and then verbally realised. Such dialogues contain both cooperative and adversarial elements, and require agents to understand, plan, and generate utterances to achieve their goals (Traum et al., 2008; Asher et al., 2012).

We collect the first large dataset of natural language negotiations between two people, and show that end-to-end neural models can be trained to negotiate by maximizing the likelihood of human actions. This approach is scalable and domain-independent, but does not model the strategic skills required for negotiating well. We further show that models can be improved by training and decoding to maximize reward instead of likelihood—by training with self-play reinforcement learning, and using rollouts to estimate the expected reward of utterances during decoding.

To study semi-cooperative dialogue, we gather a dataset of 5808 dialogues between humans on a negotiation task. Users were shown a set of items with a value for each, and asked to agree how to divide the items with another user who has a different, unseen, value function (Figure\textsuperscript{1}).

We first train recurrent neural networks to imitate human actions. We find that models trained to maximise the likelihood of human utterances can generate fluent language, but make comparatively poor negotiators, which are overly willing to compromise. We therefore explore two methods for improving the model’s strategic reasoning skills—both of which attempt to optimise for the agent’s goals, rather than simply imitating humans:

Firstly, instead of training to optimise likelihood, we show that our agents can be considerably improved using self play, in which pre-trained models practice negotiating with each other in order to optimise performance. To avoid the models diverging from human language, we interleave reinforcement learning updates with supervised updates. For the first time, we show that end-to-end dialogue agents trained using reinforcement learning outperform their supervised counterparts in negotiations with humans.

Secondly, we introduce a new form of planning for dialogue called dialogue rollouts, in which an agent simulates complete dialogues during decoding to estimate the reward of utterances. We show
that decoding to maximise the reward function (rather than likelihood) significantly improves performance against both humans and machines.

Analysing the performance of our agents, we find evidence of sophisticated negotiation strategies. For example, we find instances of the model feigning interest in a valueless issue, so that it can later ‘compromise’ by conceding it. Deceit is a complex skill that requires hypothesising the other agent’s beliefs, and is learnt relatively late in child development (Talwar and Lee, 2002). Our agents have learnt to deceive without any explicit human design, simply by trying to achieve their goals.

The rest of the paper proceeds as follows: §2 describes the collection of a large dataset of human-human negotiation dialogues. §3 describes a baseline supervised model, which we then show can be improved by goal-based training (§4) and decoding (§5). §6 measures the performance of our models and humans on this task, and §7 gives a detailed analysis and suggests future directions.

2 Data Collection

2.1 Overview

To enable end-to-end training of negotiation agents, we first develop a novel negotiation task and curate a dataset of human-human dialogues for this task. This task and dataset follow our proposed general framework for studying semi-cooperative dialogue. Initially, each agent is shown an input specifying a space of possible actions and a reward function which will score the outcome of the negotiation. Agents then sequentially take turns of either sending natural language messages, or selecting that a final decision has been reached. When one agent selects that an agreement has been made, both agents independently output what they think the agreed decision was. If conflicting decisions are made, both agents are given zero reward.

2.2 Task

Our task is an instance of multi issue bargaining (Fershtman, 1990), and is based on DeVault et al. (2015). Two agents are both shown the same collection of items, and instructed to divide them so that each item assigned to one agent.

Each agent is given a different randomly generated value function, which gives a non-negative value for each item. The value functions are constrained so that: (1) the total value for a user of all items is 10; (2) each item has non-zero value to at least one user; and (3) some items have non-zero value to both users. These constraints enforce that it is not possible for both agents to receive a maximum score, and that no item is worthless to both agents, so the negotiation will be competitive. After 10 turns, we allow agents the option to complete the negotiation with no agreement, which is worth 0 points to both users. We use 3 item types (books, hats, balls), and between 5 and 7 total items in the pool. Figure 1 shows our interface.

2.3 Data Collection

We collected a set of human-human dialogues using Amazon Mechanical Turk. Workers were paid $0.15 per dialogue, with a $0.05 bonus for maximal scores. We only used workers based in the United States with a 95% approval rating and at least 5000 previous HITs. Our data collection interface was adapted from that of Das et al. (2016).

We collected a total of 5808 dialogues, based on 2236 unique scenarios (where a scenario is the...
### 3 Likelihood Model

We propose a simple but effective baseline model for the conversational agent, in which a sequence-to-sequence model is trained to produce the complete dialogue, conditioned on an agent’s input.

#### 3.1 Data Representation

Each dialogue is converted into two training examples, showing the complete conversation from the perspective of each agent. The examples differ on their input goals, output choice, and whether utterances were read or written.

Training examples contain an input goal \( g \), specifying the available items and their values, a dialogue \( x \), and an output decision \( o \) specifying which items each agent will receive. Specifically, we represent \( g \) as a list of six integers corresponding to the count and value of each of the three item types. Dialogue \( x \) is a list of tokens \( x_{0:T} \) containing the turns of each agent interleaved with symbols marking whether a turn was written by the agent or their partner, terminating in a special token indicating one agent has marked that an agreement has been made. Output \( o \) is six integers describing how many of each of the three item types are assigned to each agent. See Figure 2.

#### 3.2 Supervised Learning

We train a sequence-to-sequence network to generate an agent’s perspective of the dialogue conditioned on the agent’s input goals (Figure 3a).

The model uses 4 recurrent neural networks, implemented as GRUs (Cho et al., 2014): GRU\(_w\), GRU\(_g\), GRU\(_\gamma\), and GRU\(_\xi\).

The agent’s input goals \( g \) are encoded using GRU\(_g\). We refer to the final hidden state as \( h^g \). The model then predicts each token \( x_t \) from left to right, conditioned on the previous tokens and \( h^g \). At each time step \( t \), GRU\(_w\) takes as input the previous hidden state \( h_{t-1} \), previous token \( x_{t-1} \) (embedded with a matrix \( E \)), and input encoding \( h^g \). Conditioning on the input at each time step helps the model learn dependencies between language and goals.

\[
    h_t = \text{GRU}_w(h_{t-1}, [E x_{t-1}, h^g]) \tag{1}
\]

The token at each time step is predicted with a softmax, which uses weight tying with the embedding matrix \( E \) (Mao et al., 2015):

\[
    p_o(x_t | x_{0:t-1}, g) \propto \exp(E^T h_t) \tag{2}
\]
Note that the model predicts both agent’s words, enabling its use as a forward model in Section 5.

At the end of the dialogue, the agent outputs a set of tokens \( o \) representing the decision. We generate each output conditionally independently, using a separate classifier for each. The classifiers share bidirectional GRUs and attention mechanism (Bahdanau et al., 2014) over the dialogue, and additionally conditions on the input goals.

\[
\begin{align*}
  h_t^\theta &= \text{GRU}_\theta(h_{t-1}^\theta, [E x_t, h_t]) \\
  h_t^\omega &= \text{GRU}_\omega(h_{t-1}^\omega, [E x_t, h_t]) \\
  h_t^o &= [h_t^\omega, h_t^\theta] \\
  h_t^\nu &= \text{tanh}(W'T'h_t^\nu) \\
  \alpha_t &= \frac{\exp(w \cdot h_t^\nu)}{\sum_{t'} \exp(w \cdot h_{t'}^\nu)} \\
  h^* &= \text{tanh}(W^s[h^\theta, \sum_t \alpha_t h_t])
\end{align*}
\]

The output tokens are predicted using softmax:

\[
p_{\theta}(o_t | x_{0..t}, g) \propto \exp(W^o h^*)
\]

The model is trained to minimize the negative log likelihood of the token sequence \( x_{0..t} \) conditioned on the input goals \( g \), and of the outputs \( o \) conditioned on \( x \) and \( g \). The two terms are weighted with a hyperparameter \( \alpha \).

\[
L(\theta) = -\sum_{x, g} \sum_t \log p_{\theta}(x_t | x_{0..t-1}, g) \\
- \alpha \sum_{x, g, o} \sum_j \log p_{\theta}(o_j | x_{0..t}, g)
\]

Unlike the Neural Conversational Model (Vinyals and Le, 2015), our approach shares all parameters for reading and generating tokens.

3.3 Decoding

During decoding, the model must generate an output token \( x_t \) conditioned on dialogue history \( x_{0..t-1} \) and input goals \( g \), by sampling from \( p_{\theta} \):

\[
x_t \sim p_{\theta}(x_t | x_{0..t-1}, g)
\]

If the model generates a special end-of-turn token, it then encodes a series of tokens output by the other agent, until its next turn (Figure 3b).

The dialogue ends when either agent outputs a special end-of-dialogue token. The model then outputs a set of choices \( o \). We choose each item independently, but enforce consistency by checking the solution is in a feasible set \( O \):

\[
o^* = \arg\max_{o \in O} \prod_t p_{\theta}(o_t | x_{0..t}, g)
\]

In our task, a solution is feasible if each item is assigned to exactly one agent. The space of solutions is small enough to be tractably enumerated.

4 Goal-based Training

Supervised learning aims to imitate the actions of human users, but does not explicitly attempt to maximise an agent’s goals. Instead, we explore pre-training with supervised learning, and then fine-tuning against the evaluation metric using reinforcement learning. Similar two-stage learning strategies have been used previously (e.g. Li et al., 2016; Das et al., 2017).

During reinforcement learning, an agent \( A \) attempts to improve its parameters from conversations with another agent \( B \). While the other agent \( B \) could be a human, in our experiments we used our fixed supervised model that was trained to imitate humans. The second model is fixed as we found that updating the parameters of both agents led to divergence from human language. In effect,
agent $A$ learns to improve by simulating conversations with the help of a surrogate forward model.

Agent $A$ reads its goals $g$ and then generates tokens $x_{0:n}$ by sampling from $p_0$. When $x$ generates an end-of-turn marker, it then reads in tokens $x_{n+1:m}$ generated by agent $B$. These turns alternate until one agent emits a token ending the dialogue. Both agents then output a decision $o$ and collect a reward from the environment (which will be 0 if they output different decisions). We denote the subset of tokens generated by $A$ as $X^A$ (e.g. tokens with incoming arrows in Figure 3b).

After a complete dialogue has been generated, we update agent $A$’s parameters based on the outcome of the negotiation. Let $r^A$ be the score agent $A$ achieved in the completed dialogue, $T$ be the length of the dialogue, $\gamma$ be a discount factor that rewards actions at the end of the dialogue more strongly, and $\mu$ be a running average of expected dialogue rewards so far. We define the future reward $R$ for an action $x_t \in X^A$ as follows:

$$R(x_t) = \sum_{x_t \in X^A} \gamma^{T-t}(r^A(o) - \mu)$$

We then optimise the expected reward of each action $x_t \in X^A$:

$$L^{RL}_\theta = \mathbb{E}_{x_t \sim p_\theta(x_t|x_{0:t-1}, g)}[R(x_t)]$$

The gradient of $L^{RL}_\theta$ is calculated as in REINFORCE (Williams, 1992):

$$\nabla_\theta L^{RL}_\theta = \sum_{x_t \in X^A} \mathbb{E}_{x_t}[R(x_t) \nabla_\theta \log(p_\theta(x_t|x_{0:t-1}, g))]$$

As all rewards are non-negative, we instead re-scale them by subtracting the mean reward found during self-play. Shifting in this way can reduce the variance of our estimator.

### 5 Goal-based Decoding

Likelihood-based decoding (13.3) may not be optimal. For instance, an agent may be choosing between accepting an offer, or making a counter offer. The former will often have a higher likelihood under our model, as there are fewer ways to agree than to make another offer, but the latter may lead to a better outcome. Goal-based decoding also allows more complex dialogue strategies. For example, a deceptive utterance is likely to have a low model score (as users were generally honest in the supervised data), but may achieve high reward.

We instead explore decoding by maximising expected reward. We achieve this by using $p_\theta$ as a

---

**Figure 4:** Decoding through rollouts: The model first generates a small set of candidate responses. For each candidate it simulates the future conversation by sampling, and estimates the expected future reward by averaging the scores. The system outputs the candidate with the highest expected reward.
forward model for the complete dialogue, and then
deterministically computing the reward. Rewards for
an utterance are averaged over samples to calcu-
late expected future reward (Figure 4).

We use a two stage process: First, we generate c candidate utterances $U = u_{0..c}$, represent-
ing possible complete turns that the agent could
make, which are generated by sampling from $p_\theta$
until the end-of-turn token is reached. Let $x_{0..n-1}$
be current dialogue history. We then calculate the
expected reward $R(u)$ of candidate utterance
$u = x_{n,n+k}$ by repeatedly sampling $x_{n+k+1..T}$
from $p_\theta$, then choosing the best output $o$ using
Equation (12), and finally deterministically comput-
ing the reward $r(o)$. The reward is scaled by the
probability of the output given the dialogue, be-
cause if the agents select different outputs then
they both receive 0 reward.

$$R(x_{n,n+k}) = \mathbb{E}_{x_{(n+k+1..T,o)}} p_\theta[r(o)p_\theta(o|x_{0..T})]$$

We then return the utterance maximizing $R$.

$$u^* = \arg\max_{u \in U} R(u)$$

We use 5 rollouts for each of 10 candidate turns.

6 Experiments

6.1 Training Details

We implement our models using PyTorch. All
hyper-parameters were chosen on a development
dataset. The input tokens are embedded into a 64-di-
ensional space, while the dialogue tokens are
embedded with 256-dimensional embeddings
(with no pre-training). The input $GRU_g$ has a
hidden layer of size 64 and the dialogue $GRU_w$
is of size 128. The output $GRU_g$ and $GRU_w$
both have a hidden state of size 256, the size of $h^g$
is 256 as well. During supervised training, we
optimise using stochastic gradient descent with a
 minibatch size of 16, an initial learning rate of
1.0, Nesterov momentum with $\mu = 0.1$ [Nesterov,
1983], and clipping gradients whose $L^2$ norm ex-
ceeds 0.5. We train the model for 30 epochs and
pick the snapshot of the model with the best val-
idation perplexity. We then annealed the learn-
ing rate by a factor of 5 each epoch. We weight
the terms in the loss function (Equation 10) using
$\alpha = 0.5$. We do not train against output decisions
where humans selected different agreements. To-
kens occurring fewer than 20 times are replaced
with an ‘unknown’ token.

During reinforcement learning, we use a learn-
ing rate of 0.1, clip gradients above 1.0, and use
a discount factor of $\gamma = 0.95$. After every 4 rein-
forcement learning updates, we make a supervised
update with mini-batch size 16 and learning rate
0.5, and we clip gradients at 1.0. We used 4086
simulated conversations.

When sampling words from $p_\theta$, we reduce the
variance by doubling the values of logits (i.e. us-
ting temperature of 0.5).

6.2 Comparison Systems

We compare the performance of the following:
LIKELIHOOD uses supervised training and decod-
ing (13), RL is fine-tuned with goal-based self-
play (14), ROLLOUTS uses supervised training
combined with goal-based decoding using rollouts
(15), and RL+ROLLOUTS uses rollouts with a base
model trained with reinforcement learning.

6.3 Intrinsic Evaluation

For development, we use measured the perplexity
of user generated utterances, conditioned on the
input and previous dialogue.

Results are shown in Table 3 and show that
the simple LIKELIHOOD model produces the most
human-like responses, and the alternative training
and decoding strategies cause a divergence from
human language. Note however, that this diver-
gence may not necessarily correspond to lower
quality language—it may also indicate different
strategic decisions about what to say. Results in
6.4 show all models could converse with humans.

6.4 End-to-End Evaluation

We measure end-to-end performance in dialogues
both with the likelihood-based agent and with hu-
mans on Mechanical Turk, on held out scenarios.

Humans were told that they were interacting
with other humans, as they had been during the
collection of our dataset (and few appeared to re-
alize they were in conversation with machines).

We measure the following statistics:
Score: The average score for each agent (which
could be a human or model), out of 10.
Agreement: The percentage of dialogues where
both agents agreed on the same decision.
Pareto Optimality: The percentage of Pareto
optimal solutions for agreed deals (a solution is
Pareto optimal if neither agent’s score can be im-
proved without lowering the other’s score). Lower
scores indicate inefficient negotiations.
### Table 1: End task evaluation on heldout scenarios, against the LIKELIHOOD model and humans from Mechanical Turk. The maximum score is 10. Score (all) gives 0 points when agents failed to agree.

| Model            | Score (all) | Score (agreed) | % Agreed | % Pareto Optimal |
|------------------|-------------|----------------|----------|------------------|
| LIKELIHOOD       | 5.4 vs. 5.5 | 6.2 vs. 6.2    | 87.9     | 49.6             |
| RL               | 7.1 vs. 4.2 | 7.9 vs. 4.7    | 89.9     | 58.6             |
| ROLLOUTS         | 7.3 vs. 5.1 | 7.9 vs. 5.5    | 92.9     | 63.7             |
| RL+ROLLOUTS      | 8.3 vs. 4.2 | 8.8 vs. 4.5    | 94.4     | 74.8             |
| Score (agreed)   | 4.7 vs. 5.8 | 6.2 vs. 7.6    | 76.5     | 66.2             |
| % Agreed         | 4.3 vs. 5.0 | 6.4 vs. 7.5    | 67.3     | 69.1             |
| % Pareto Optimal | 5.2 vs. 5.4 | 7.1 vs. 7.4    | 72.1     | 78.3             |
| Score (agreed)   | 4.6 vs. 4.2 | 8.0 vs. 7.1    | 57.2     | 82.4             |

### Table 2: Statistics on our dataset of crowdsourced dialogues between humans.

| Metric                        | Dataset |
|-------------------------------|---------|
| Number of Dialogues           | 5808    |
| Average Turns per Dialogue    | 6.6     |
| Average Words per Turn        | 7.6     |
| % Agreed                      | 80.1    |
| Average Score (/10)           | 6.0     |
| % Pareto Optimal              | 76.9    |

### Table 3: Intrinsic evaluation showing the average perplexity of tokens and rank of complete turns (out of 2083 unique human messages from the test set). Lower is more human-like for both.

| Model            | Valid PPL | Test PPL | Test Avg. Rank |
|------------------|-----------|----------|----------------|
| LIKELIHOOD       | 5.62      | 5.47     | 521.8          |
| RL               | 6.03      | 5.86     | 517.6          |
| ROLLOUTS         | -         | -        | 844.1          |
| RL+ROLLOUTS      | -         | -        | 859.8          |

7 Analysis

Table 1 shows large gains from goal-based methods. In this section, we explore the strengths and weaknesses of our models.

**Goal-based models negotiate harder.** The RL+ROLLOUTS model has much longer dialogues with humans than LIKELIHOOD (7.2 turns vs. 5.3 on average), indicating that the model is accepting deals less quickly, and negotiating harder.

A negative consequence of this more aggressive negotiation strategy is that humans were more likely to walk away with no deal, which is reflected in the lower agreement rates. Even though failing to agree was worth 0 points, people often preferred this course over capitulating to an uncompromising opponent—a factor not well captured by the simulated partner in reinforcement learning training or rollouts (as reflected by the larger gains from goal-based models in dialogues with the LIKELIHOOD model). In particular, the goal-based models are prone to simply rephrasing the same demand each turn, which is a more effective strategy against the LIKELIHOOD model than humans. Future work should address this issue.

Figure 5 shows an example of our goal-based model stubbornly negotiating until it achieves a good outcome.

**Models learn to be deceptive.** Deception can be an effective negotiation tactic. We found numerous cases of our models initially feigning interest in a valueless item, only to later ‘compromise’ by conceding it. Figure 7 shows an example.

**Models produce meaningful novel sentences.** One interesting question is whether our models are capable of generating novel sentences in the new circumstances they find themselves in, or if they simply repeat messages from the training data verbatim. We find that 76% of messages produced by the LIKELIHOOD model in self-play were found in the training data. We manually examined the novel
has assumed that state representations are anno-
ted in the training data (Williams and Young, 2007; Henderson et al., 2014; Wen et al., 2016).
The use of state annotations allows a cleaner separation of the reasoning and natural language aspects of dialogues, but our end-to-end approach makes data collection cheaper and allows tasks where it is unclear how to annotate state. Bordes and Weston (2016) explore end-to-end goal orientated dialogue with a supervised model—we show improvements over supervised learning with goal-based training and decoding. Recently, He et al. (2017) use task-specific rules to combine the task input and dialogue history into a more structured state representation than ours.

Reinforcement learning (RL) has been applied in many dialogue settings. RL has been widely used to improve dialogue managers, which manage transitions between dialogue states (Singh et al., 2002; Pietquin et al., 2011; Rieser and Lemon, 2011; Gašic et al., 2013; Fatemi et al., 2016). In contrast, our end-to-end approach has no explicit dialogue manager. Li et al. (2016) improve metrics such as diversity for non-goal-orientated dialogue using RL, which would make an interesting extension to our work. Das et al. (2017) use reinforcement learning to improve cooperative bot-bot dialogues. RL has also been used to allow agents to invent new languages (Das et al., 2017; Mordatch and Abbeel, 2017). To our knowledge, our model is the first to use RL to improve the performance of an end-to-end goal orientated dialogue system in dialogues with humans.

Work on learning end-to-end dialogues has concentrated on ‘chat’ settings, without explicit goals (Ritter et al., 2011; Vinyals and Le, 2015; Li et al., 2015). These dialogues contain a much greater diversity of vocabulary than our domain, but do not

| Input | RL+ROLLOUTS | Human |
|-------|-------------|-------|
| book 3 | book value=3 | book value=3 |
| hat 1 | hat value=1 | hat value=1 |
| ball 2 | ball value=2 | ball value=1 |

| Output | Reward |
|--------|--------|
| book 3 | 10/10 |
| hat 1 | 6/10 |

Figure 5: Example of a model stubbornly holding out for maximum points in a dialogue.

| Input | RL+ROLLOUTS | Human |
|-------|-------------|-------|
| book 4 | book value=4 | book value=2 |
| hat 5 | hat value=5 | hat value=1 |
| ball 0 | ball value=0 | ball value=2 |

| Output | Reward |
|--------|--------|
| hat 5 | 10/10 |
| book 4 | 6/10 |

Figure 7: Dialogue in which the model’s initial interest in the valueless books allows it to compromise while achieving a maximum score.

utterances produced by our model, and found that the overwhelming majority were fluent English sentences in isolation—showing that the model has learnt a good language model for the domain (in addition to results that show it uses language effectively to achieve its goals). These results suggest that although neural models are prone to the safer option of repeating sentences from training data, they are capable of generalising when necessary. Future work should choose domains that force a higher degree of diversity in utterances.

Maintaining multi-sentence coherence is challenging. One common linguistic error we see RL+ROLLOUTS make is to start a message by indicating agreement (e.g. I agree or Deal), but then going on to propose a counter offer—a behaviour that human partners found frustrating. One explanation is that the model has learnt that in the supervised data, messages beginning with I agree are often at the end of the dialogue, and partners rarely reply with further negotiation—so the models using rollouts and reinforcement learning believe this tactic will help their offer to be accepted.

8 Related Work

Most work on goal orientated dialogue systems has assumed that state representations are anno-

Figure 6: Example of model compromising.
have the challenging adversarial elements. Such models are notoriously hard to evaluate (Liu et al., 2016), because the huge diversity of reasonable responses, whereas our task has a clear objective. Our end-to-end approach would also be much more straightforward to integrate into a general-purpose dialogue agent than one that relied on annotated dialogue states (Dodge et al., 2016).

There is a substantial literature on multi-agent bargaining in game-theory, e.g. Nash Jr (1950). There has also been computational work on modelling negotiations (Baarslag et al., 2013)—our work differs in that agents communicate in unrestricted natural language, rather than pre-specified symbolic actions, and our focus on improving performance relative to humans rather than other automated systems. Our task is based on that of DeVault et al. (2015), who study natural language negotiations for pedagogical purposes—their version includes speech rather than textual dialogue, and embodied agents, which would make interesting extensions to our work. The only automated natural language negotiations systems we are aware of have first mapped language to domain-specific logical forms, and then focused on choosing the next dialogue act (Rosenfeld et al., 2014; Cuayahuitl et al., 2015; Keizer et al., 2017). Our end-to-end approach is the first to to learn comprehension, reasoning and generation skills in a domain-independent data driven way.

Our use of a combination of supervised and reinforcement learning for training, and stochastic rollouts for decoding, builds on strategies used in game playing agents such as AlphaGo (Silver et al., 2016). Our work is a step towards real-world applications for these techniques. Our use of rollouts could be extended by choosing the other agent’s responses based on sampling, using Monte Carlo Tree Search (MCTS) (Kocsis and Szepesvári, 2006). However, our setting has a higher branching factor than in domains where MCTS has been successfully applied, such as Go (Silver et al., 2016)—future work should explore scaling tree search to dialogue modelling.

9 Conclusion

We have introduced end-to-end learning of natural language negotiations as a task for AI, arguing that it challenges both linguistic and reasoning skills while having robust evaluation metrics. We gathered a large dataset of human-human negotiations, which contain a variety of interesting tactics. We have shown that it is possible to train dialogue agents end-to-end, but that their ability can be much improved by training and decoding to maximise their goals, rather than likelihood. There remains much potential for future work, particularly in exploring other reasoning strategies, and in improving the diversity of utterances without diverging from human language. We will also explore other negotiation tasks, to investigate whether models can learn to share negotiation strategies across domains.

Acknowledgments

We would like to thank Luke Zettlemoyer and the anonymous EMNLP reviewers for their insightful comments, and the Mechanical Turk workers who helped us collect data.

References

Nicholas Asher, Alex Lascarides, Oliver Lemon, Markus Guhe, Verena Rieser, Philippe Muller, Stergos Afantinos, Farah Benamara, Laure Vieu, Pascal Denis, et al. 2012. Modelling Strategic Conversation: The STAC project. Proceedings of SemDial page 27.

Tim Baarslag, Katsuhide Fujita, Enrico H Gerding, Koen Hindriks, Takayuki Ito, Nicholas R Jennings, Catholijn Jonker, Sarit Kraus, Raz Lin, Valentin Robu, et al. 2013. Evaluating Practical Negotiating Agents: Results and Analysis of the 2011 International Competition. Artificial Intelligence 198:73–103.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv preprint arXiv:1409.0473.

Antoine Bordes and Jason Weston. 2016. Learning End-to-End Goal-oriented Dialog. arXiv preprint arXiv:1605.07683.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

Heriberto Cuayahuitl, Simon Keizer, and Oliver Lemon. 2015. Strategic Dialogue Management via Deep Reinforcement Learning. arXiv preprint arXiv:1511.08099.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2016. Visual Dialog. arXiv preprint arXiv:1611.08669.
Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. 2017. Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning. arXiv preprint arXiv:1703.06585.

David DeVault, Johnathan Mell, and Jonathan Gratch. 2015. Toward Natural Turn-taking in a Virtual Human Negotiation Agent. In AAAI Spring Symposium on Turn-taking and Coordination in Human-Machine Interaction. AAAI Press, Stanford, CA.

Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander H. Miller, Arthur Szlam, and Jason Weston. 2016. Evaluating Pre-requisite Qualities for Learning End-to-End Dialog Systems. arXiv preprint arXiv:1606.03152.

Chaim Fershtman. 1990. The Importance of the Agenda in Bargaining. Games and Economic Behavior 2(3):224–238.

Mihdi Fatemi, Layla El Asri, Hannes Schulz, Jing He, and Kaheer Suleman. 2016. Policy Networks with Two-stage Training for Dialogue Systems. arXiv preprint arXiv:1606.03152.

H. He, A. Balakrishnan, M. Eric, and P. Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In Association for Computational Linguistics (ACL).

Matthew Henderson, Blaise Thomson, and Jason Williams. 2014. The Second Dialog State Tracking Challenge. In 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue. volume 263.

Simon Keizer, Markus Guhe, Heriberto Cuayáhuithil, Ioannis Efthiathou, Klaus-Peter Engelbrecht, Mihai Dobre, Alexandra Lascarides, and Oliver Lemon. 2017. Evaluating Persuasion Strategies and Deep Reinforcement Learning methods for Negotiation Dialogue agents. In Proceedings of the European Chapter of the Association for Computational Linguistics (EACL 2017).

Levente Kocsis and Csaba Szepesvári. 2006. Bandit based Monte-Carlo Planning. In European conference on machine learning. Springer, pages 282–293.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A Diversity-promoting Objective Function for Neural Conversation Models. arXiv preprint arXiv:1510.03055.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016. Deep Reinforcement Learning for Dialogue Generation. arXiv preprint arXiv:1606.01541.

Chia-Wei Liu, Ryan Lowe, Iulian V. Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing.

Junhua Mao, Xu Wei, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan L. Yuille. 2015. Learning Like a Child: Fast Novel Visual Concept Learning From Sentence Descriptions of Images. In The IEEE International Conference on Computer Vision (ICCV).

Igor Mordatch and Pieter Abbeel. 2017. Emergence of Grounded Compositional Language in Multi-Agent Populations. arXiv preprint arXiv:1703.04908.

John F. Nash Jr. 1950. The Bargaining Problem. Econometrica: Journal of the Econometric Society pages 155–162.

Yurii Nesterov. 1983. A Method of Solving a Convex Programming Problem with Convergence Rate O (1/k2). In Soviet Mathematics Doklady. volume 27, pages 372–376.

Olivier Pietquin, Matthieu Geist, Senthilkumar Chandramohan, and Hervé Fressa-Buet. 2011. Sample-efficient Batch Reinforcement Learning for Dialogue Management Optimization. ACM Trans. Speech Lang. Process. 7(3):7:1–7:21.

Verena Rieser and Oliver Lemon. 2011. Reinforcement Learning for Adaptive Dialogue Systems: A Data-driven Methodology for Dialogue Management and Natural Language Generation. Springer Science & Business Media.

Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven Response Generation in Social Media. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pages 583–593.

Avi Rosenfeld, Inon Zuckerman, Erel Segal-Halevi, Osnat Drein, and Sarit Kraus. 2014. NegoChat: A Chat-based Negotiation Agent. In Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, AAMAS ’14, pages 525–532.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the Game of Go with Deep Neural Networks and Tree Search. Nature 529(7587):484–489.

Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. 2002. Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System. Journal of Artificial Intelligence Research 16:105–133.
Victoria Talwar and Kang Lee. 2002. Development of lying to conceal a transgression: Children’s control of expressive behaviour during verbal deception. *International Journal of Behavioral Development* 26(5):436–444.

David Traum, Stacy C. Marsella, Jonathan Gratch, Jina Lee, and Arno Hartholt. 2008. Multi-party, Multi-issue, Multi-strategy Negotiation for Multi-modal Virtual Agents. In *Proceedings of the 8th International Conference on Intelligent Virtual Agents*, Springer-Verlag, Berlin, Heidelberg, IVA ’08, pages 117–130.

Oriol Vinyals and Quoc Le. 2015. A Neural Conversational Model. *arXiv preprint arXiv:1506.05869*.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A Network-based End-to-End Trainable Task-oriented Dialogue System. *arXiv preprint arXiv:1604.04562*.

Jason D Williams and Steve Young. 2007. Partially Observable Markov Decision Processes for Spoken Dialog Systems. *Computer Speech & Language* 21(2):393–422.

Ronald J Williams. 1992. Simple Statistical Gradient-following Algorithms for Connectionist Reinforcement Learning. *Machine learning* 8(3-4):229–256.