Recommendation as a Communication Game: Self-Supervised Bot-Play for Goal-oriented Dialogue

Dongyeop Kang[♡] **Anusha Balakrishnan**[♣] **Pararth Shah**[♣]
Paul Crook[♣] **Y-Lan Boureau**[♣] **Jason Weston**[♣]
[♡]Carnegie Mellon University, [♣]Facebook AI
dongyeok@cs.cmu.edu anusha.bala28@gmail.com
{pararths,pacrook,ylan,jase}@fb.com

Abstract

Traditional recommendation systems produce static rather than interactive recommendations invariant to a user's specific requests, clarifications, or current mood, and can suffer from the cold-start problem if their tastes are unknown. These issues can be alleviated by treating recommendation as an interactive dialogue task instead, where an expert recommender can sequentially ask about someone's preferences, react to their requests, and recommend more appropriate items. In this work, we collect a goal-driven recommendation dialogue dataset (GoRecDial), which consists of 9,125 dialogue games and 81,260 conversation turns between pairs of human workers recommending movies to each other. The task is specifically designed as a cooperative game between two players working towards a quantifiable common goal. We leverage the dataset to develop an end-to-end dialogue system that can simultaneously converse and recommend. Models are first trained to imitate the behavior of human players without considering the task goal itself (supervised training). We then fine-tune our models on simulated bot-bot conversations between two paired pre-trained models (bot-play), in order to achieve the dialogue goal. Our experiments show that models fine-tuned with bot-play learn improved dialogue strategies, reach the dialogue goal more often when paired with a human, and are rated as more consistent by humans compared to models trained without bot-play. The dataset and code are publicly available through the ParlAI framework[1].

1 Introduction

Traditional recommendation systems factorize users' historical data (i.e., ratings on movies) to extract common preference patterns (Koren et al., 2009; He et al., 2017b). However, besides making it difficult to accommodate new users because of the *cold-start problem*, relying on aggregated history makes these systems static, and prevents users from making specific requests, or exploring a temporary interest. For example, a user who usually likes horror movies, but is in the mood for a fantasy movie, has no way to indicate their preference to the system, and would likely get a recommendation that is not useful. Further, they cannot iterate upon initial recommendations with clarifications or modified requests, all of which are best specified in natural language.

Recommending through dialogue interactions (Reschke et al., 2013; Wärnestål, 2005) offers a promising solution to these problems, and recent work by Li et al. (2018) explores this approach in detail. However, the dataset introduced in that work does not capture higher-level strategic behaviors that can impact the quality of the recommendation made (for example, it may be better to elicit user preferences first, before making a recommendation). This makes it difficult for models trained on this data to learn optimal recommendation strategies. Additionally, the recommendations are not grounded in real observed movie preferences, which may make trained models less consistent with actual users. This paper aims to provide *goal-driven recommendation dialogues grounded in real-world data*. We collect a corpus of goal-driven dialogues grounded in real user movie preferences through a carefully designed gamified setup (see Figure 1) and show that models trained with that corpus can learn a successful recommendation dialogue strategy. The training is conducted in two stages: first, a *supervised* phase that trains the model to mimic human behavior on the task; second, a *bot-play* phase that improves the goal-directed strategy of the model.

The contribution of this work is thus twofold.

[1] https://github.com/facebookresearch/ParlAI

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 1951–1961, Hong Kong, China, November 3–7, 2019.

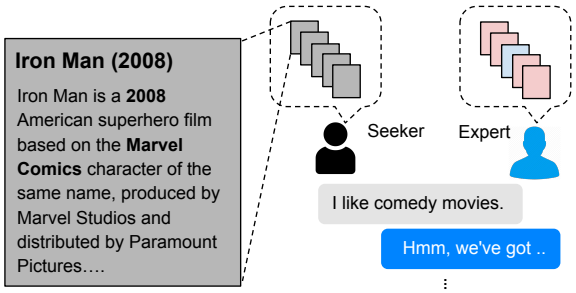


Figure 1: Recommendation as a dialogue game. We collect 81,260 recommendation utterances between pairs of human players (experts and seekers) with a collaborative goal: the expert must recommend the correct (blue) movie, avoiding incorrect (red) ones, and the seeker must accept it. A chatbot is then trained to play the expert in the game.

(1) We provide the first (to the best of our knowledge) large-scale goal-driven recommendation dialogue dataset with specific goals and reward signals, grounded in a real-world knowledge base. (2) We propose a two-stage recommendation strategy learning framework and empirically validate that it leads to better recommendation conversation strategies.

2 Recommendation Dialogue Task Design

In this section, we first describe the motivation and design of the dialogue-based recommendation game that we created. We then describe the data collection environment and present detailed dataset statistics.

2.1 Dialogue Game: Expert and Seeker

The game is set up as a conversation between a *seeker* looking for a movie recommendation, and an *expert* recommending movies to the seeker. Figure 2 shows an example movie recommendation dialogue between two-paired human workers on Amazon Mechanical Turk.

Game Setting. Each worker is given a set of five movies[2] with a description (first paragraph from the Wikipedia page for the movie) including important features such as director's name, year, and genre. The seeker's set represents their watching history (movies they are supposed to have liked) for the game's sake. The expert's set consists of candidate movies to choose from when making recommendations, among which only one is the *correct* movie to recommend. The correct movie is chosen to be similar to the seeker's movie set (see Sec. 2.2), while the other four movies are dissimilar. The expert is not told by the system which of the five movies is the correct one. The expert's goal is to find the correct movie by chatting with the seeker and recommend it after a minimal number of dialogue turns. The seeker's goal is to accept or reject the recommendation from the expert based on whether they judge it to be similar to their set. The game ends when the expert has recommended the correct movie. The system then asks each player to rate the other for engagingness.

Justification. Players are asked to provide reasons for recommending, accepting, or rejecting a movie, so as to get insight into human recommendation strategies[3].

Gamification. Rewards and penalties are provided to players according to their decisions, to make the task more engaging and incentivize better strategies. Bonus money is given if the expert recommends the correct movie, or if the seeker accepts the correct movie or rejects an incorrect one.

2.2 Picking Expert and Seeker movie sets

This section describes how movie sets are selected for experts and seekers.

Pool of movies To reflect movie preferences of real users, our dataset uses the MovieLens dataset[4], comprising 27M ratings applied to 58K movies by 280K real users. We obtain descriptive text for each movie from Wikipedia[5] (i.e., the first paragraph). We also extract entity-level features (e.g., directors, actors, year) using the MovieWiki dataset (Miller et al., 2016) (See Figure 1). We filter out less frequent movies and user profiles (see Appendix), resulting in a set of 5,330 movies and 65,181 user profiles with their ratings.

Movie similarity metric In order to simulate a natural setting, the movies in the seeker's set

[2]We deliberately restricted the set of movies to make the task more tractable. One may argue that the expert can simply ask these candidates one by one (at the cost of low engagingness). However, this empirically doesn't happen: experts make on average only 1.16 incorrect movie recommendations.

[3]Our model doesn't utilize this or the engagingness scores for learning, but these are potential future directions.

[4]https://grouplens.org/datasets/movielens/

[5]https://dumps.wikimedia.org/

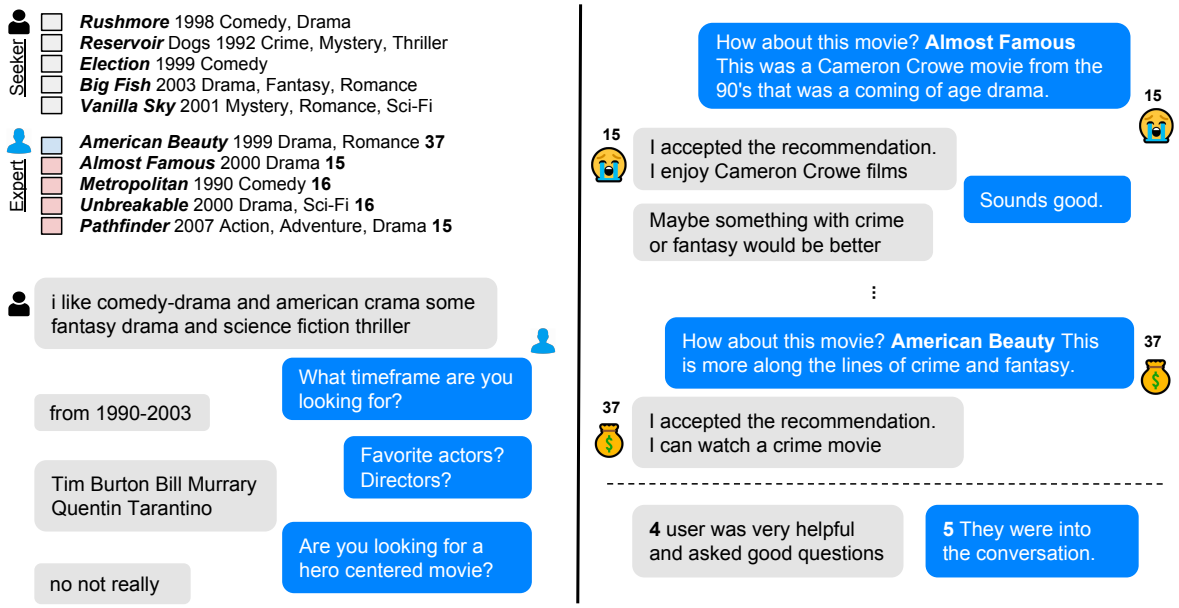


Figure 2: An example dialogue from our dataset of movie recommendation between two human workers: seeker (grey) and expert (blue). The goal is for the expert to find and recommend the correct movie (light blue) out of incorrect movies (light red) which is similar to the seeker movies. Best viewed in color.

should be similar to each other, and the correct movie should be similar to these, according to a metric that reflects coherent empirical preferences. To compute such a metric, we train an embedding-driven recommendation model (Wu et al., 2018).[6] Each movie is represented as an embedding, which is trained so that embeddings of movies watched by the same user are close to each other. The closeness metric between two movies is the cosine similarity of these trained embeddings. A movie is deemed close to a set of movies if its embedding is similar to the average of the movie embeddings in the set.

Movie Set Selection Using these trained embeddings, we design seeker and expert sets based on the following criteria (See Figure 3):

- Seeker movies (grey) are a set of five movies which are close to each other, chosen from the set of all movies watched by a real user.
- The correct movie (light blue) is close to the average of the five embeddings of the seeker set.
- The expert's incorrect movies (light red) are far from the seeker set and the correct movie.

We filter out movie sets that are too difficult or easy for the recommendation task (see Appendix), and choose 10,000 pairs of seeker-expert movie sets at random.

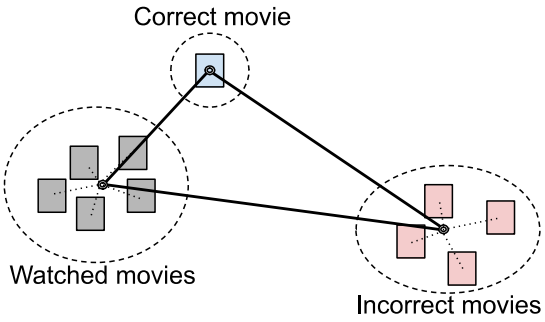


Figure 3: Movie set selection: watched movies for seeker (grey) and correct (light blue) / incorrect (light red) movies for expert.

2.3 Data Collection

For each dialogue game, a movie set is randomly chosen without duplication. We collect dialogues using ParlAI (Miller et al., 2017) to interface with Amazon Mechanical Turk. More details about data collection are included in the Appendix.

Table 1 shows detailed statistics of our dataset regarding the movie sets, the annotated dialogues, actions made by expert and seeker, dialogue

[6]We also tried a classical matrix-factorization based recommendation model, which shows comparable performance to the embedding model.

Dialogue statistics	
Number of dialogues	9,125
Number of utterances	170,904
Number of unique utterances	85,208
Avg length of a dialogue	23.0
Avg duration (minutes) of a dialogue	5.2
Expert's utterance statistics	
Avg utterance length	8.40
Unique tokens	11,757
Unique utterances	40,550
Seeker's utterance statistics	
Avg utterance length	8.47
Unique tokens	10,766
Unique utterances	45,196
Action statistics (all scores are averaged)	
# of correct/incorrect recs. by expert	1.0 / 1.16
# of correct/incorrect decisions by seeker	1.1 / 1.04
Game statistics (all scores are averaged)	
min/max movie scores	12.3 / 46.0
correct/incorrect movies	39.9 / 15.0
real game score by expert/seeker	61.3 / 50.8
random game score by expert/seeker	43.2 / 38.1
Engagingness statistics (all scores are averaged)	
engagingness score by expert/seeker	4.3 / 4.4
engagingness scores & feedback collected	18,308

Table 1: Data statistics. "correct/incorrect" in the action stats means that the expert recommends the correct/incorrect movie or the seeker correctly accepts/rejects the movie.

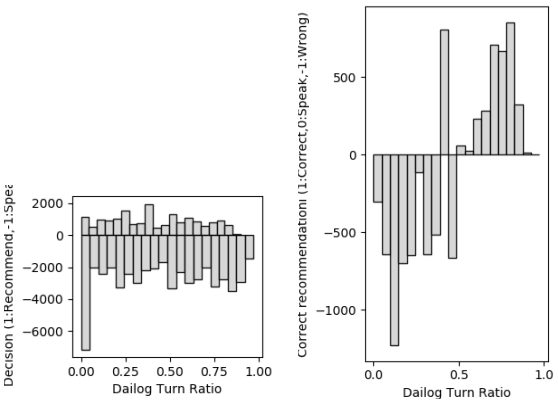
games, and engagingness feedback.

The collected dialogues contain a wide variety of action sequences (recommendations and accept/reject decisions). Experts make an average of 1.16 incorrect recommendations, which indicates a reasonable difficulty level. Only 37.6% of dialogue games end at first recommendation, and 19.0% and 10.8% at second and third recommendations, respectively.

Figure 4 shows histogram distributions of (a) expert's decisions between speaking utterance and recommendation utterance and (b) correct and incorrect recommendations over the normalized turns of dialogue. In (a), recommendations increasingly occur after a sufficient number of speaking utterances. In (b), incorrect recommendations are much more frequent earlier in the dialogue, while the opposite is true later on.

3 Our Approach

In order to recommend the right movie in the role of the expert, a model needs to combine several perceptual and decision skills. We propose to conduct learning in two stages (See Figure 5): *supervised multi-aspect learning* and *bot-play*.



(a) Decision to speak (-1) or recommend (+1)

(b) Correct (+1) or incorrect (-1) recommendations

Figure 4: Histogram distribution of (a) experts' decisions of whether to speak or recommend and (b) correct/incorrect recommendations over the normalized dialogue turns.

3.1 Supervised Multi-Aspect Learning

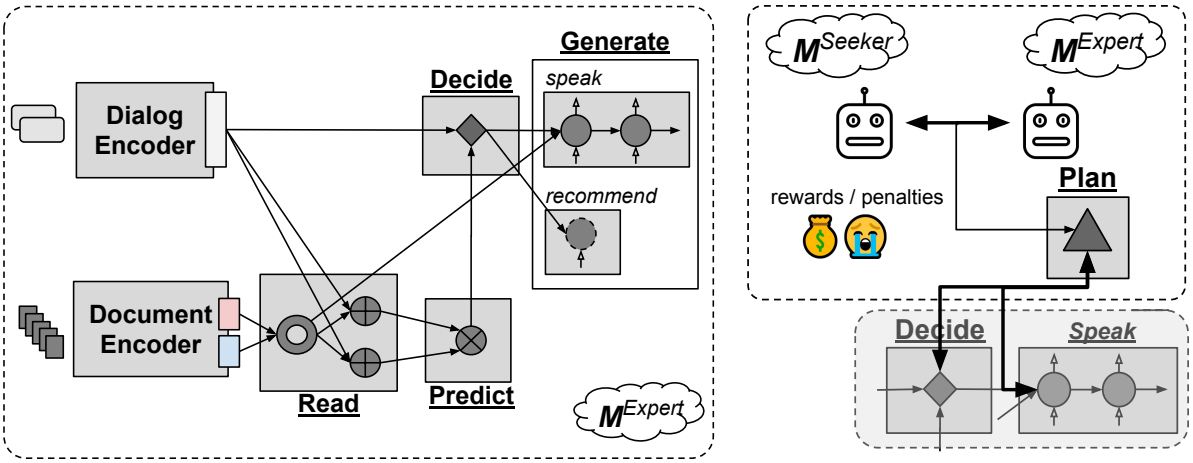
The supervised stage of training the expert model combines three sources of supervision, corresponding to the three following subtasks: (1) GENERATE dialogue utterances to speak with the seeker in a way that matches the utterances of the human speaker, (2) PREDICT the correct movie based on the dialogue history and the movie description representations, and (3) DECIDE whether to recommend or speak in a way that matches the observed decision of the human expert.

Using an LSTM-based model (Hochreiter and Schmidhuber, 1997), we represent the dialogue history context h_t of utterances x_1 to x_t as the average of LSTM representations of $x_1, \cdots, x_t$, and the description m_k of the k-th movie as the average of the bag-of-word representations[7] of its description sentences. Let (x_{t+1}, y, d_{t+1}) denote the ground truth next utterance, correct movie index, and ground truth decision at time $t+1$, respectively. We cast the supervised problem as an end-to-end optimization of the following loss:

$$\mathcal{L}_{sup} = \alpha\mathcal{L}_{gen} + \beta\mathcal{L}_{predict} + (1-\alpha-\beta)\mathcal{L}_{decide}, \quad (1)$$

where α and β are weight hyperparameters optimized over the validation set, and $\mathcal{L}_{predict}, \mathcal{L}_{decide}, \mathcal{L}_{gen}$ are negative log-likelihoods

[7] We empirically found that BOW works better than other encoders such as LSTM in this case.



(a) Supervised multi-aspect learning

(b) Bot-play

Figure 5: (a) Supervised learning of the expert model $\mathcal{M}^{expert}$ and (b) bot-play game between the expert $\mathcal{M}^{expert}$ and the seeker $\mathcal{M}^{seeker}$ models. The former imitates multiple aspects of humans' behaviors in the task, while the later fine-tunes the expert model w.r.t the game goal (i.e., recommending the correct movie).

of probability distributions matching each of the three subtasks:

$$\mathcal{L}_{gen} = -\log p_{gen}(x_{t+1}|h_t, m_1, \cdots, m_K), \quad (2)$$

$$\mathcal{L}_{predict} = -\log p(y|c_1, \cdots, c_K), \quad \text{where} \quad (3)$$

$$c_j = h_t \cdot m_j \qquad \text{for } j \in 1..K, \quad (4)$$

$$\mathcal{L}_{decide} = p_{MLP}(d_{t+1}|h_t, c_1, \cdots, c_K), \quad (5)$$

with p_{gen} the output distribution of an attentive seq2seq generative model (Bahdanau et al., 2015), p a softmax distribution over dot products $h_t \cdot m_k$ that capture how aligned the dialogue history h_t is with the description m_k of the k-th movie, and p_{MLP} the output distribution of a multi-layer perceptron predictor that takes $c_1, \cdots, c_K$ as inputs[8].

3.2 Bot-Play

Motivated by the recent success of self-play in strategic games (Silver et al., 2017; Vinyals et al., 2019; OpenAI, 2018) and in negotiation dialogues (Lewis et al., 2017), we show in this section how we construct a reward function to perform bot-play between two bots in our setting, with the aim of developing a better expert dialogue agent for recommendation.

Plan optimizes long-term policies of the various aspects over multiple turns of the dialogue game by maximizing game-specific rewards. We first pre-train expert and seeker models individually: the expert model $\mathcal{M}^{expert}(\theta) = \min_\theta \mathcal{L}_{sup}$ is pre-trained by minimizing the supervised loss in Eq 1, and the seeker model $\mathcal{M}^{seeker}(\phi)$ is a retrieval-based model that retrieves seeker utterances from the training set based on cosine similarity of the preceding dialogue contexts encoded using the BERT pre-trained encoder[9]. θ and ϕ are model parameters of the expert and seeker model, respectively. Then, we make them chat with each other, and fine-tune the expert model by maximizing its reward in the game (See Figure 5, Right).

The dialogue game ends if the expert model recommends the correct movie, or a maximum dialogue length is reached[10], yielding T turns of dialogue; $g = (x_1^{expert}, x_1^{seeker} .. x_T^{expert}, x_T^{seeker})$. Let T_{REC} the set of turns when the expert made a recommendation. We define the expert's reward as:

$$r_t^{expert} = \frac{1}{|T_{REC}|} \cdot \sum_{t \in T_{REC}} \delta^{t-1} \cdot b_t, \quad (6)$$

where δ is a discount factor[11] to encourage earlier recommendations, b_t is the reward obtained at each recommendation made, and $|T_{REC}|$ is the number of recommendations made. b_t is 0 unless the correct movie was recommended.

[8] We experimented with various other encoding functions, detailed in the Appendix.

[9] See Sec. 4.2 for details on BERT. We also experimented with sequence-to-sequence models for modeling the seeker but performance was much worse.

[10] We restrict the maximum length of a dialogue to 20.

[11] we use $\delta = 0.5$.

We define the reward function $\mathcal{R}$ as follows:

$$\mathcal{R}(x_t) = \sum_{x_t \in X^{expert}} \gamma^{T-t}(r_t^{expert} - \mu) \quad (7)$$

where $\mu = \frac{1}{t}\sum_{1..t} r_t^{expert}$ is the average of the rewards received by the expert until time t and γ is a discount factor to diminish the reward of earlier actions. We optimize the expected reward for each turn of dialogue x_t and calculate its gradient using REINFORCE (Williams, 1992). The final role-playing objective $\mathcal{L}_{RP}$ is:

$$\nabla \mathcal{L}_{RP}(\theta; z) = \sum_{x_t \in X^{expert}} \mathbb{E}_{x_t}[\nabla \log p(x_t|x_{<t})\mathcal{R}(x_t)] \quad (8)$$

We optimize the role-playing objective with the pre-trained expert model's decision ($\mathcal{L}_{decide}$) and generation ($\mathcal{L}_{gen}$) objectives at the same time. To control the variance of the RL loss, we alternate optimizing the RL loss and other two supervised losses for each step. We do not fine-tune the prediction loss, in order not to degrade the prediction performance during bot-play.

4 Experiments

We describe our experimental setup in §4.1. We then evaluate our supervised and unsupervised models in §4.2 and §4.3, respectively.

4.1 Setup

We select 5% of the training corpus as validation set in our training.

All hyper-parameters are chosen by sweeping different combinations and choosing the ones that perform best on the validation set. In the following, the values used for the sweep are given in brackets. Tokens of textual inputs are lower-cased and tokenized using byte-pair-encoding (BPE) (Sennrich et al., 2016) or the Spacy[12] tokenizer. The seq-to-seq model uses 300-dimensional word embeddings initialized with GloVe (Pennington et al., 2014) or Fasttext (Joulin et al., 2017) embeddings, $[1, 2]$ layers of $[256, 512]$-dimensional Uni/Bi-directional LSTMs (Hochreiter and Schmidhuber, 1997) with 0.1 dropout ratio, and soft attention (Bahdanau et al., 2015). At decoding, we use beam search with a beam of size 3, and choose the maximum likelihood output. For each turn, the initial movie text and all previous dialogue turns including seeker's and expert's replies are concatenated as input to the models.

Both supervised and bot-play learning use Adam (Kingma and Ba, 2015) optimizer with batch size 32 and learning rates of $[0.1, 0.01, 0.001]$ with 0.1 gradient clipping. The number of softmax layers (Yang et al., 2018) is $[1, 2]$. For each turn, the initial movie description and all previous dialogue utterances from the seeker and the expert are concatenated as input text to the other modules. Each movie textual description is truncated at 50 words for efficient memory computation.

We use annealing to balance the different supervised objectives: we only optimize the GENERATE loss for the first 5 epochs, and then gradually increase weights for the PREDICT and DECIDE losses. We use the same movie-sets as in the supervised phase to fine-tune the expert model. Our models are implemented using PyTorch and ParlAI (Miller et al., 2017). Code and dataset will be made publicly available through ParlAI[13].

4.2 Evaluation of Supervised Models

Metrics. We first evaluate our supervised models on the three supervised tasks: dialogue generation, movie recommendation, and per-turn decision to speak or recommend. The dialogue generation is evaluated using the F1 score and BLEU (Papineni et al., 2002) comparing the predicted and ground-truth utterances. The F1 score is computed at token-level. The recommendation model is evaluated by calculating the percentage of times the correct movie is among the top k recommendations (hit@k). In order to see the usefulness of dialogue for recommendation, precision is measured per each expert turn of the dialogue (`Turn@k`) regardless of the decision to speak or recommend, and at the end of the dialogue (`Chat@k`).

Models. We compare our models with Information Retrieval (IR) based models and recommendation-only models. The IR models retrieve the most relevant utterances from the set of candidate responses of the training data and rank them by comparing cosine similarities using TFIDF features or BERT (Devlin et al., 2019) encoder features. Note that IR models make no recommendation. The recommendation-only models

[12] https://spacy.io/

[13] https://github.com/facebookresearch/ParlAI

		Generation		Recommendation				Decision
		F1	BLEU	Turn@1	Turn@3	Chat@1	Chat@3	Acc
Baseline	TFIDF-RANKER	32.5	**27.8**	-	-	-	-	-
	BERT-RANKER	38.3	23.9	-	-	-	-	
	RANDOM RECC.	3.6	0.1	21.3	59.2	23.1	62.2	-
	BERT RECC.	16.5	0.2	25.5	66.3	26.4	68.3	-
Ours	GENERATE	39.5	26.0	-	-	-	-	-
	+PREDICT	40.2	26.4	76.4	96.9	75.7	97.0	-
	+DECIDE	**41.0**	27.4	**77.8**	**97.1**	**78.2**	**97.7**	**67.6**
	+PLAN	40.9	26.8	76.3	95.7	77.5	97.6	53.6

Table 2: Evaluation on supervised models. We incrementally add different aspects of modules: GENERATE, PREDICT, and DECIDE for supervised multi-aspect learning and PLAN for bot-play fine-tuning.

always produce recommendation utterances following the template (e.g., "how about this movie, [MOVIE]?") where the [MOVIE] is chosen randomly or based on cosine similarities between dialogue contexts and the text descriptions of candidate movies. We use the pre-trained BERT encoder (Devlin et al., 2019) to encode dialogue contexts and movie text descriptions.

We incrementally add each module to our base GENERATE model: PREDICT and DECIDE for supervised learning and PLAN for bot-play fine-tuning. Each model is chosen from the best model in our hyper-parameter sweeping.

Results. Table 2 shows performance comparison on the test set. Note that only the full supervised model (+DECIDE) and the fine-tuned model (+PLAN) can appropriately operate every function required of an expert agent such as producing utterances, recommending items, and deciding to speak or recommend.

Compared to recommendation-only models, our prediction PREDICT modules show significant improvements over the recommendation baselines on both per-turn and per-chat recommendations: 52% on Turn@1 and 34% on Turn@3. Chat scores are always higher than Turn, indicating that recommendations get better as more dialogue context is provided. The DECIDE module yields additional improvements over the PREDICT model in both generation and recommendation, with 67.6% decision accuracy, suggesting that the supervised signal of decisions to speak or recommend can contribute to better overall representations.

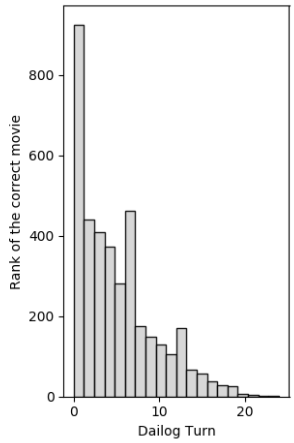
In generation, our proposed models show comparable performance as the IR baseline models (e.g., BERTRANKER). The +DECIDE model improves on the F1 generation score because it learns when to predict the templated recommendation utterance.

As expected, +PLAN slightly hurts most metrics of supervised evaluation, because it optimizes a different objective (the game objective), which might not systematically align with the supervised metrics. For example, a system optimized to maximize game objective should try to avoid incorrect recommendations even if humans made them. Game-related evaluations are shown in §4.3.

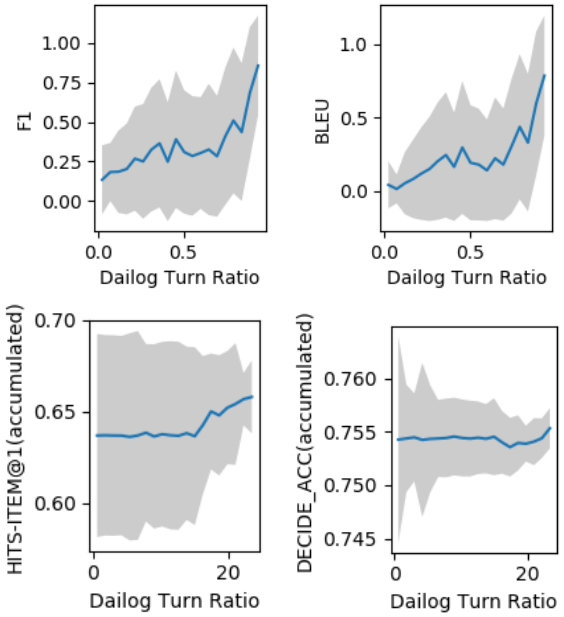
Analysis We analyze how each of the supervised modules acts over the dialogue turns on the test set. Figure 6 (a) shows a histogram of the rank of the ground-truth movie over turns. The rank of the model's prediction is very high for the first few turns, then steadily decreases as more utterances are exchanged with the seeker. This indicates that the dialogue context is crucial for finding good recommendations.

The evolution of generation metrics (F1, BLEU) for each turn is shown at the top of Fig. 6(b), and the (accumulated) recommendation and decision metrics (Turn@1/Accuracy) at the bottom[14]. The accumulated recommendation and decision performance sharply rises at the end of the dialogue and variance decreases. The generation performance increases, because longer dialogue contexts helps predict the correct utterances.

[14]For better understanding of the effect of recommendation and decision, we show accumulated values, and per-turn values for generation.



(a) Rank of recommendation



(b) F1/BLEU (top) & Turn@1 /Decision Acc (bottom)

Figure 6: Analysis of the expert's model: as the dialogue continues (x-axis is either fraction of the full dialogue, or index of dialogue turn), y-axis is (a) rank of the correct recommendation (the lower rank, the better) and (b) F1/BLEU/Turn@1/Decision Accuracy (the higher the better) with the variance shown in grey.

4.3 Evaluation on Dialogue Games

Metrics. In the bot-play setting, we provide game-specific measures as well as human evaluations. We use three automatic game measures: Goal to measure the ratio of dialogue games where the goal is achieved (i.e., recommending the correct movie or not), Score to measure the total game score, and Turn2G to count the number of dialogue turns taken until the goal is achieved.

We conduct human evaluation by making the expert model play with human seekers. We measure automatic metrics as well as dialogue quality scores provided by the player: fluency, consistency, and engagingness (scored between 1 and 5) (Zhang et al., 2018). We use the full test set (i.e., 911 movie sets) for bot-bot games and use 20 random samples from the test set for {bot,human}-human games.

Models. We compare our best supervised model with several variants of our fine-tuned bot-play models. We consider bot-play of an expert model with different seeker models such as BERT-Ranker based seeker and Seq-to-Seq based seeker. Each bot-play model is trained on the same train set that is used for training the original supervised model. The seeker model uses retrieval based on BERT pretrained representations of dialogue context (BERT-R) [15].

Players		Automatic			Human		
Expert	**Seeker**	**Goal**	**Sco**	**T2G**	**F**	**C**	**E**
Supervised*	BERT-R	30.9	38.3	1.4	-	-	-
Bot-play \w S2S	BERT-R	42.1	49.6	2.8	-	-	-
Bot-play \w BERT-R	BERT-R	**48.6**	**52.4**	**3.2**	-	-	-
Supervised*	Human	55.0	51.2	2.1	3.1	2.2	**2.0**
Bot-play*	Human	**68.5**	**54.7**	**3.1**	**3.2**	**2.6**	**2.0**
Human	Human	95.0	64.3	8.5	4.8	4.7	4.2

Table 3: Evaluation on dialogue recommendation games: bot-bot (top three rows) and {bot,human}-human (bottom three rows). We use automatic game measures (**Goal**, **Score**, **Turn2Goal**) and human quality ratings (**Fluency**, **Consistency**, **Engagingness**).

Results. Compared to the supervised model, the self-supervised model fine-tuned by seeker models shows significant improvements in the game-related measures. In particular, the BERT-R model shows a +27.7% improvement in goal success ratio. Interestingly, the number of turns to reach the goal increases from 1.4 to 3.2, indicating that conducting longer dialogues seems to be a better strategy to achieve the game goal throughout our role-playing game.

In dialogue games with human seeker players, the bot-play model also outperforms the supervised one, even though it is still far behind human performance. When the expert bot plays with the human seeker, performance increases compared to

[15] A potential direction for future work may have more solid seeker models and explore which aspect of the model makes the dialogue with the expert model more goal-oriented or human-like.

playing with the bot seeker, because the human seeker produces utterances more relevant to their movie preferences, increasing overall game success.

5 Related Work

Recommendation systems often rely on matrix factorization (Koren et al., 2009; He et al., 2017b). Content (Mooney and Roy, 2000) and social relationship features (Ma et al., 2011) have also been used to help with the cold-starting problem of new users. The idea of eliciting users' preference for certain content features through dialogue has led to several works. Wärneståal (2005) studies requirements for developing a conversational recommender system, e.g., accumulation of knowledge about user preferences and database content. Reschke et al. (2013) automatically produces template-based questions from user reviews. However, no conversational recommender systems have been built based on these works due to the lack of a large publicly available corpus of human recommendation behaviors.

Very recently, Li et al. (2018) collected the ReDial dataset, comprising 10K conversations of movie recommendations, and used it to train a generative encoder-decoder dialogue system. In this work, crowdsource workers freely talk about movies and are instructed to make a few movie recommendations before accepting one. Compared to ReDial, our dataset is grounded in real movie preferences (movie ratings from MovieLens), instead of relying on workers' hidden movie tastes. This allows us to make our task goal-directed rather than chit-chat; we can optimize prediction and recommendation strategy based on a known ground truth, and train the predict and plan modules of our system. That in turn allows for novel setups such as bot-play.

To the best of our knowledge, Bordes et al. (2016) is the only other goal-oriented dialogue benchmark grounded in a database that has been released with a large-scale publicly available dataset. Compared to that work, our database is made of real (not made-up) movies, and the choice of target movies is based on empirical distances between movies and movie features instead of being arbitrary. This, combined with the collaborative set-up, makes it possible to train a model for the seeker in the bot-play setting.

Our recommendation dialogue game is collaborative. Other dialogue settings with shared objectives have been explored, for example a collaborative graph prediction task (He et al., 2017a), and semi-cooperative negotiation tasks (Lewis et al., 2017; Yarats and Lewis, 2018; He et al., 2018).

6 Conclusion and Future Directions

In conclusion, we have posed recommendation as a goal-oriented game between an expert and a seeker, and provided a framework for both training agents in a supervised way by learning to mimic a large set of collected human-human dialogues, as well as by bot-play between trained agents. We have shown that a combination of the two stages leads to learning better expert recommenders.

Our results suggest several promising directions. First, we noted that the recommendation performance linearly increases as more dialogue context is provided. An interesting question is how to learn to produce the best questions that will result in the most informative dialogue context.

Second, as the model becomes better at the game, we observe an increase in the length of dialogue. However, it remains shorter than the average length of human dialogues, possibly because our reward function is designed to minimize it, which worked better in experiments. A potential direction for future work is to study how different game objectives interact with each other.

Finally, our evaluation on movie recommendation is made only within the candidate set of movies given to expert. Future work should evaluate if our training scheme generalizes to a fully open-ended recommendation system, thus making our task not only useful for research and model development, but a useful end-product in itself.

Acknowledgements

We thank Eduard Hovy, Alan W Black, Dan Jurafsky, Alan Ritter, and anonymous reviewers for their helpful comments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017a. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *ACL*.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343, Brussels, Belgium. Association for Computational Linguistics.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017b. Neural collaborative filtering. In *WWW*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *EACL*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42.

Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning for negotiation dialogues. In *EMNLP*.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *NIPS*, pages 9725–9735.

Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *WSDM*.

Kanti V Mardia. 1970. Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57(3):519–530.

Alexander H. Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. 2017. Parlai: A dialog research software platform. In *EMNLP*.

Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *EMNLP*.

Raymond J Mooney and Loriene Roy. 2000. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204. ACM.

OpenAI. 2018. Openai five. https://blog.openai.com/openai-five/.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Kevin Reschke, Adam Vogel, and Daniel Jurafsky. 2013. Generating recommendation dialogs by extracting information from user reviews. In *ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *Nature*, 550(7676):354.

Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojciech M. Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, Timo Ewalds, Dan Horgan, Manuel Kroiss, Ivo Danihelka, John Agapiou, Junhyuk Oh, Valentin Dalibard, David Choi, Laurent Sifre, Yury Sulsky, Sasha Vezhnevets, James Molloy, Trevor Cai, David Budden, Tom Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Toby Pohlen, Yuhuai Wu, Dani Yogatama, Julia Cohen, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Chris Apps, Koray Kavukcuoglu, Demis Hassabis, and David Silver. 2019. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/.

Pontus Wärnestål. 2005. Modeling a dialogue strategy for personalized movie recommendations. In *Beyond Personalization Workshop*, pages 77–82.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Ledell Yu Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. 2018. Starspace: Embed all the things! In *AAAI*.

Zhilin Yang, Zihang Dai, Ruslan R. Salakhutdinov, and William W. Cohen. 2018. Breaking the softmax bottleneck: A high-rank rnn language model. In *ICLR*.

Denis Yarats and Mike Lewis. 2018. Hierarchical text generation and planning for strategic dialogue. In *ICML*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *ACL*.