Strategy and Policy Learning for Non-Task-Oriented Conversational Systems

Zhou Yu, Ziyu Xu, Alan W Black and Alex I. Rudnicky
School of Computer Science
Carnegie Mellon University
{zhouyu,awb,air}@cs.cmu.edu, ziyux@andrew.cmu.edu

Abstract

We propose a set of generic conversational strategies to handle possible system breakdowns in non-task-oriented dialog systems. We also design dialog policies to select among these strategies with respect to different dialog contexts. We combine expert knowledge and the statistical findings that derived from previous collected data in designing these policies. The dialog policy learned via reinforcement learning outperforms the random selection policy and the locally greedy policy in both the simulated and the real-world settings. In addition, we propose three metrics, which consider both the local and global quality of the conversation, to evaluate conversation quality.

1 Introduction

There are a variety of methods to generate responses for non-task-oriented systems, such as machine translation (Ritter et al., 2011), retrieval-based response selection (Banchs and Li, 2012), and sequence-to-sequence recurrent neural network (Vinyals and Le, 2015). However, these systems still produce utterances that are incoherent or inappropriate from time to time. To tackle this problem, we propose a set of conversational strategies, such as switching topics, to avoid possible inappropriate responses (breakdowns). Another difficulty is to decide which strategy to select with respect to different conversational contexts. In a multi-turn conversation, if the same strategy is used repeatedly, the user experience will be affected. We experiment on three dialog policies: a random selection policy that randomly selects a policy regardless of the dialog context, a locally greedy policy that focuses on local dialog context, and a reinforcement learning policy that considers the entire dialog context. The conversational strategies and policies are applicable for non-task-oriented systems in general, regardless of the response generation method. The conversational strategies can prevent a possible breakdown. The probability of possible breakdowns can be calculated using different metrics according to different systems. For example, a neural network generation system (Vinyals and Le, 2015) can use the posterior probability to decide if the generated utterance would cause a system breakdown or not. We implement a set of conversational strategies and three policies in a keyword retrieval non-task-oriented system. We use the retrieval confidence as the criteria to decide whether a strategy is needed to be triggered to avoid system breakdowns.

Reinforcement learning was introduced to the dialog community two decades ago (Biermann and Long, 1996) and has mainly been used in task-oriented systems (Singh et al., 1999). Researchers have proposed to design dialogue systems in the formalism of Markov decision processes (MDPs) (Levin et al., 1997) or partially observable Markov decision processes (POMDPs) (Williams and Young, 2007). In a stochastic environment, a dialog system's actions are system utterances, and the state is represented by the dialog history. The goal is to design a dialog system that takes actions to maximize some measure of system reward, such as task completion rate or dialog length. The difficulty of such modeling lies in the state representation. Representing the dialog by the entire history is often neither feasible nor conceptually useful, and the so-called belief state approach is not possible, since we do not even know what features are required to represent the belief state. Previous work (Walker et al., 1998) has largely dealt with this issue by imposing prior limitations on the features used to represent the approximate state. In this paper, instead of focusing on task-oriented systems, we apply reinforcement learning to design a policy to select designed conversation strategies in a non-task-oriented dialog systems. Unlike task-oriented dialog systems, non-task-oriented systems have no specific goal that guides the interaction. Consequently, evaluation metrics that are traditionally used for reward design, such as task completion rate, are no longer appropriate. The state design in reinforcement learning is even more difficult for non-task-oriented systems, as the same conversation would not occur more than once; one slightly different answer would lead to a completely different conversation; moreover there is no clear sense of when such a conversation is "complete". We simplify the state design by introducing expert knowledge, such as not repeating the same strategy in a row, as well as statistics obtained from conversational data analysis.

We implement and deploy a non-task-oriented dialog system driven by a statistical policy to avoid possible system breakdowns using designed conversation strategies. We evaluate the system on the Amazon Mechanical Turk platform with metrics that consider both the local and the global quality of the conversation.

2 Related Work

Many generic conversational strategies have been proposed in previous work to avoid generating incoherent utterances in non-task-oriented conversations, such as introducing new topics (e.g. "Let's talk about favorite foods!") in (Higashinaka et al., 2014), asking the user to explain missing words (e.g. "What is SIGDIAL?") (Maria Schmidt and Waibel, 2015). We propose a set of generic strategies that are inspired by previous work, and test their usability on human users. No researcher has investigated thoroughly on which strategy to use in different conversational contexts. Compared to task-oriented dialog systems, non-task-oriented systems have more varied conversation history, which are thus harder to formulate as a mathematical problem. In this work, we propose a method to use statistical findings in conversational study to constrain the dialog history space and to use reinforcement learning for statistical policy learning in a non-task-oriented conversation setting.

To date, reinforcement learning is mainly used for learning dialogue policies for slot-filling task-oriented applications such as bus information search (Lee and Eskenazi, 2012), restaurant recommendations (Jurčíček et al., 2012), and sightseeing recommendations (Misu et al., 2010). Reinforcement learning is also used for some more complex systems, such as learning negotiation policies (Georgila and Traum, 2011) and tutoring (Chi et al., 2011). Reinforcement learning is also used in question-answering systems (Misu et al., 2012). Question-answering systems are very similar to non-task-oriented systems except that they do not consider dialog context in generating responses. They have pre-existing questions that the user is expected to go through, which limits the content space of the dialog. Reinforcement learning has also been applied to a non-task-oriented system for deciding which sub-system to choose to generate a system utterance (Shibata et al., 2014). In this paper, we used reinforcement learning to learn a policy to sequentially decide which conversational strategy to use to avoid possible system breakdowns.

Task completion rate is widely used as the conversational metric for task oriented systems (Williams and Young, 2007). However, it is not applicable for non-task-oriented dialog systems, as they don't have a task. Response appropriateness (coherence) is a widely used manual annotation metric (Yu et al., 2016) for non-task-oriented systems. However, this metric only focuses on the utterance level conversational quality and is not automatically computable. Perplexity of the language model is an automatically computable metric but is difficult to interpret (Vinyals and Le, 2015). We propose three metrics: *turn-level appropriateness*, *conversational depth* and *information gain*, which assess both the local and the global conversation quality. Although only *Information gain* is automatically quantifiable, we use supervised machine learning methods to built automatic detectors for *turn level appropriateness* and *conversational depth.*

3 Conversational Strategy Design

We design three types of conversational strategies: context tracking strategies, lexical semantic strategies and general diversion strategies, to improve TickTock's response appropriateness. We first randomly sample 10% of the conversations generated using Text-TickTock 1.0 and classify the breakdowns of the system (turns that are rated "Inappropriate" or "Interpretable" based on Table 1) into different types and evaluate them on the data collected by Text-TickTock 2.0. In Text-TickTock 2.0, we apply all context tracking strategies to all user utterances. If the retrieval confidence score is high, we use the retrieved response directly. If the retrieval confidence score is low, we first test if lexical semantic strategies are applicable. If none of them is applicable, we randomly select one of the general diversion strategies. In Figure 1, we illustrate how the strategies are applied in sequence.

3.1 Context Tracking Strategies

We design two conversational strategies to incorporate history information of the conversation to generate responses.

1. **Anaphora Resolution.** We find that users use a lot of pronouns when talking to the chatbot in the previous chapter. An example input would be "I hate them" and here "then" refers back to the topic, ''sports" in the previous conversational turn. Anaphora detection is difficult when the sentence structure is complex. Luckily in the everyday chatting conversation, the sentence structure is relative simple. By substituting the pronoun by the noun of the previous sentence already covers 85% of the cases. Thus we only implement this simple rule to resolve anaphoras. This case triggered 30 times in the TickTock 2.0 generated conversations.

2. **Response Ranking with History Similarities.** We first retrieve five candidate responses from the database using keyword retrieval methods. Then we adjust the rank of these candidates based on how similar the candidate is to the previous user response. We compute the two utterances response using word2vec (Mikolov et al., 2013).

3.2 Lexical Semantic Strategies

We design four lexical semantic strategies that utilize the lexical and semantic information to deal with cases that no similar sentences can be found in the chatbot database.

1. **Don't repeat.** If the user repeat themselves, the system confronts the user by saying:"You already said that!". It triggered 5 times in the Text-TickTock 2.0 generated conversations and all of them are rated by the users as "Appropriate".

2. **Ground on named entities.** We perform a shallow parsing to find the named entity in the sentence, and then retrieve a short description of the named entity in a knowledge base. Finally we use several templates to generate sentences using the obtained short description. One example reply is "Are you talking about Chicago, the city in Illinois?" This strategy is considered a type of grounding strategy in human conversations. Users feel like they are understood when this strategy is triggered correctly. In addition, we make sure we never ground the same named-entity twice in single conversation. It triggered 22 times in the Text-TickTock 2.0 generated conversations and 92% (20 out of 22) of the times, users rated the system responses as "Appropriate". The cases that are rated "Inappropriate" are mainly caused by the errors generated by named entity detection.

3. **Ground on out of vocabulary words.** If the user says a word that is out of the system's vocabulary, such as "confrontational". Then the chatbot asks: "What is confrontational?" We expand our vocabulary with the new user-defined words continuously, so we will not ask for grounding on the same word twice. It triggered 36 times in the Text-TickTock 2.0 generated conversations and 83% (30 out of 36) of the time the users rated the generated response as "Appropriate". The ones that are rated "Interpretable" and "Inappropriate" are mainly caused by using this strategy in sequence and the users find the response weird once the system does it too often.

4. **React to single-word sentence.** If user types in meaningless single word such as 'd', 'dd', or equations such as '1+2='. Then the chatbot replies: "Can you be serious and say things in a complete sentence?" to deal with such condition. It triggered 12 times in the TickTock 2.0 generated conversations, among them all the users rated the generated responses "Appropriate".

3.3 General Diversion Strategies

We design five general diversion strategies to avoid system breakdowns when the retrieval confidence score is low and none of the semantic lexical strategies are applicable.

Label	Definition	Example
Inappropriate (Inapp)	Not coherent with the user utterance	*Participant*: How old are you? *TickTock*: Apple.
Interpretable (Inter)	Related and can be interpreted	*Participant*: How old are you? *TickTock*: That's too big a question for me to answer.
Appropriate (App)	Coherent with the user utterance	*Participant*: How is the weather today? *TickTock*: Very good.

Table 1: Appropriateness rating scheme.

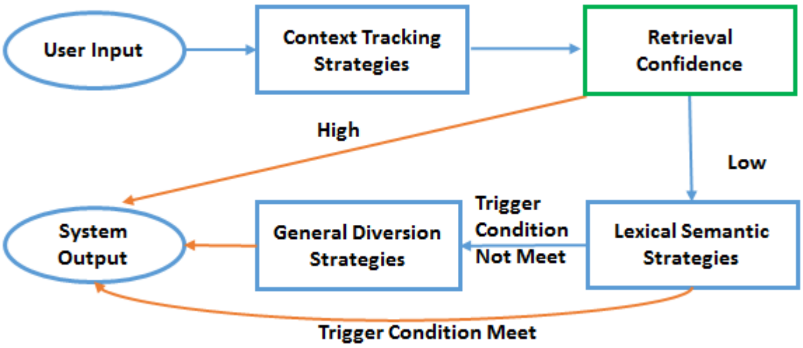
User Input
Context Tracking Strategies
Retrieval Confidence
High
Low
System Output
General Diversion Strategies
Trigger Condition Not Meet
Lexical Semantic Strategies
Trigger Condition Meet

Figure 1: System information flow diagram

1. **Switch a topic.** The chatbot proposes a new topic, such as "sports", other than the current topic. For example: "Let's talk about sports." If this strategy is executed, the system updates the tracked topic.

2. **Initiate activities.** The chatbot invites the user to an activity. Each invitation is designed to match the conversation topic. For example, under the politics topic, the system would ask: "Do you want to see the latest Star Wars movie together?"

3. **End topics with an open question.** The chatbot closes the current conversation topic and asks an open question, such as " Sorry I don't know. Could you tell me something interesting?".

4. **Tell a joke.** The chatbot tells a joke under the conversation topic, such as: "Politicians and diapers have one thing in common. They should both be changed regularly, and for the same reason".

5. **Elicit more information.** The chatbot asks the user to say more about the current topic, such as " Could we talk more about that?".

4 Dialog Policy Design

In Text-TickTock 2.0, we use a random selection policy that randomly chooses among general diversion strategies whenever lexical semantic strategies are not applicable. We find that the sentiment polarity of the utterance has an influence on which general diversion strategy to select that leads to appropriate response. People tend to rate the *switch* strategy more favorably if there is negative sentiment in the previous utterances. For example:

TickTock: Hello, I really like politics. Let's talk about politics.
User: No, I don't like politics.
TickTock: Why is that?
User: I just don't like politics.
TickTock: OK, how about we talk about movies?

In another scenario, when all the previous three utterances are positive, the *more* strategy (e.g. Do you want to talk more about that?) is preferred over the *switch* strategy (e.g. Do you like movies?).

We set out to find the optimum strategy to deal with the user utterance given the sentiment polarities of its previous three utterances. We generate five different versions of the conversations by replacing the original used general diversion strategy with other general diversion strategies. We ask people to rate the strategy's appropriateness given its three previous utterances. For each conversation, we collect ratings from three different raters and use the majority rating as the final score. Then we construct a table of a distribution that represents the system response's appropriateness regarding each strategy. We collect 10 ratings for each strategy under each context. We use the Vader (Hutto and Gilbert, 2014) sentiment predictor for automatic sentiment prediction. The sentiment predictor produces a label with three categories: positive (pos), negative(neg) and neutral (neu).

We find that the results of the rating task supports our hypothesis that different strategies are preferred with respect to different sentiment context. In Table 2, we show the distribution of the appropriateness ratings for all the general diversion strategies in a context when all their previous utterances are positive. Users rated the *more* strategy more appropriate than the *end* strategy and the *switch* strategy. One interesting observation is that the *joke* strategy is rated poorly. We examine all the cases and find that the low appropriateness rate is mostly due to the fact that the joke is unexpected given the context. The *initiation* strategy can be appropriate when the activity fits the previous content semantically.

In another sentiment context, when there are consecutive negative utterances, the *switch* strategy and the *end* strategy are preferred. We can see that which strategy is appropriate is heavily dependent on the immediately sentiment context of the conversation. Sentiment polarity captures some conversational level information which is a discriminating factor. We use these findings to design the locally greedy policy. The system deal with user's utterance uses the strategy that is rated as the most appropriate given the utterance's three previous utterances sentiment polarity .

We conduct another Amazon Mechanical Turk study to test if sentiment context beyond three utterances would influence the preferred strategy or not. To reduce the work load, we test on one condition which is when the previous three utterances are all positive. We provide the complete conversation history of that dialog to the raters instead of only three previous utterances. We find that strategies used most recently are rated less favorably if used again. This motivates us to include information that relates to the usage of the previous strategy and a longer history to design policy that cares about global context.

Strategy	App	Inter	Inapp
switch	0.1	0.3	0.6
initiation	0.2	0.4	0.4
joke	0.1	0.2	0.7
end	0.1	0.3	0.6
more	0.4	0.5	0.1

Table 2: Appropriateness rating distribution when the recent three utterances are positive.

5 Reinforcement Learning

We model the conversation process as a Markov Decision Process (MDP)-based problem, so we can use reinforcement learning to learn a conversational policy that makes sequential decisions by considering the entire context. We used Q-learning, a model-free method to learn the conversational policy for our non-task-oriented conversational system.

In reinforcement learning, the problem is defined as $(S, A, R, \gamma, \alpha)$, where S is the set of states that represents the system's environment, in this case the conversational context. A is a set of actions available per state. In our setting, the actions are strategies available. By performing an action, the agent can move from one state to another. Executing an action in a specific state provides the agent with a reward (a numerical score), $R(s, a)$. The goal of the agent is to maximize its total reward. It does this by learning which action is optimal to take for each state. The action that is optimal for each state is the action that has the highest long-term reward. This reward is a weighted sum of the expected values of the rewards of all future steps starting from the current state, where the discount factor γ is a number between 0 and 1 that trades off the importance of sooner versus later rewards. γ may also be interpreted as the likelihood to succeed (or survive) at every step. The algorithm therefore has a function that calculates the quantity of a state-action combination, $Q : S \times A \rightarrow R$. The core of the algorithm is a simple value iteration update. It assumes the old value and makes a correction based on the new information at each time step, t. See Equation (1) for details of the iteration function.

The critical part of the modeling is to design appropriate states and the corresponding reward

$$Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \alpha_t(s_t, a_t) \cdot \Big(R_{t+1} + \gamma \max_a Q_t(s_{t+1}, a) - Q_t(s_t, a_t)\Big) \tag{1}$$

$$\text{Turn-level appropriateness} * 10 + \text{Conversational depth} * 100 + round(\text{Information gain}, 5) * 30 \tag{2}$$

function. We reduce the number of the states by incorporating expert knowledge and the statistical findings in our analysis. We use another chatbot, A.L.I.C.E. [1] as a user simulator in the training process. We include features: turn index, times each strategy was executed previously, and the sentiment polarity of previous three utterances. We construct the reward table based on the statistics collected from the previous experiment. To make the reward table tractable, we impose some of the rules we constructed based on expert knowledge. For example, if certain strategy has been used before, then the reward of using it again is reduced. If the trigger condition of any lexical semantic strategies are met, the system chooses them over all general diversion strategies. This may result in some less optimum solutions, but reduces the state space and action space considerably. During the training process, we constrain the conversation to be 10 turns. The reward function is only given at the end of the conversation, which is a combination of the automatic predictions of the three metrics that consider the conversation quality both locally and globally. We will discuss them in detail in the next section. It takes 5000 conversations for the algorithm to converge. We look into the learned Q table and find that the policy prefers the strategy that uses less frequently if the context is fixed, which resonant with our previous findings. We name the version using the reinforcement learning policy as Text-TickTock 3.0.

6 Evaluation Metrics

In the learning process of the reinforcement learning, we use a metric which is a combination of three metrics: *turn-level appropriateness*, *conversational depth* and *information gain*. *Conversational depth* and *information gain* measure the quality of the conversation across multiple turns. Since we use another chatbot as the simulator, making sure the overall conversation quality is good is critical. All three metrics are related to each other but cover different aspects of the conversation. We use a weighted score of the three metrics for the learning process, which is shown in Equation (2). The coefficients are chosen based on empirical heuristics. We build automatic predictors for *turn-level appropriateness* and *conversation depth* based on annotated data as well.

6.1 Turn-Level Appropriateness

Turn-level appropriateness reflects the coherence of the system's response in each conversational turn. See Table 1 for the annotation scheme. The inter-annotator agreement between the two experts is relatively high (kappa = 0.73). We collapse the "Appropriate" and "Interpretable" labels into one class and formulate the appropriateness detection as a binary classification problem. Our designed policies and strategies intend to avoid system breakdowns (the inappropriate responses), so we built this detector to tell whether a system response is appropriate or not.

We annotated the appropriateness for 1256 turns. We balance the ratings by generating more inappropriate examples by randomly pairing two utterances. In order to reduce the variance of the detector, we use five-fold cross-validation and a Z-score normalizer to scale all the features into the same range. We use early fusion, which simply concatenates all feature vectors. We use a v-Support Vector (Chang and Lin, 2011) with a RBF Kernel to train the detector. The performance of the automatic appropriateness detector is 0.73 in accuracy while the accuracy of the majority vote is 0.5.

We use three sets of features: the strategy used in the response, the word counts of both the user's and TickTock's utterances, and the utterance similarity features. The utterance similarity features consist of a feature vector obtained from a word2vec model (Mikolov et al., 2013), the cosine similarity score between the user utterance and the system response, and the similarity scores between the user response and all the previous system responses. For the word2vec model, we trained a 100-dimension model using the collected data.

---

[1] http://alice.pandorabots.com/

Turn	Participant	TickTock
1	I am not sure. What's on your mind?	Well, you know, I'm not sure. Let me tell you something, politicians and diapers have one thing in common. They should both be changed regularly, and for the same reason.
2	Wow. That's funny and insightful.	Uhh, I can't answer this. Can we talk about something else?
3	How about movies. Have you seen any good movies?	Of course. What do you think?
4	I love super hero movies. Have you seen Deadpool?	Are you talking about Deadpool,the animation character?
5	Yes, exactly. They made a movie of him.	That sounds really interesting.

Table 3: An example conversation with TickTock in the real-world setting

Turn	A.L.I.C.E.	TickTock
1	When do I do it?	I don't know.
2	That is interesting.	Do you want to go to my friend's party tonight?
3	Sure, just download me to your notebook or PDA and take me along.	Sounds great. Do you want to play board games together some time.
4	I like to play the Turing Game.	I don't know how to play it. Is it difficult to play?
5	Interesting gossip. It could be.	Possibly.

Table 4: An example conversation of TickTock in the simulated setting

Conversational depth	Consecutive utterances
Shallow	< 6
Intermediate	[7, 10]
Deep	> 10

Table 5: Conversational depth annotation scheme

6.2 Conversational Depth

Conversational depth reflects the number of consecutive utterances that share the same topic. We design an annotation scheme (Table 5) based on the maximum number of consecutive utterances on the same topic. We annotate conversations into three categories: "Shallow", "Intermediate" and "Deep". The annotation agreement between the two experts is moderate (kappa = 0.45). Users manually labeled 100 conversations collected using Text-TickTock 1.0 and 2.0. We collapse "Shallow" and "Intermediate" into one category and formulate the problem as a binary classification problem. We use the same machine learning setting as the turn level appropriateness predictor. The performance of the automatic conversational depth detector is 72.7% accuracy, while the majority vote baseline accuracy is 63.6%. The conversational depth detector has three types of features:

1. The number of dialogue exchanges between the user and TickTock, and the number of times TickTock uses the *continue*, *switch* and *end* strategy.

2. The count of a set of keywords used in the conversation. The keywords are "sense", "something" and interrogative pronouns, such as "when", "who", "why", etc. "Sense" often occurs in sentence, such as "You are not making any sense" and "something" often occurs in sentence, such as "Can we talk about something else?" or "Tell me something you are interested in." Both of them indicate a possible topic change. Interrogative pronouns are usually involved in questions that probe users to express more on the current topic.

3. We convert the entire conversation into a vector using doc2vec and also include the cosine similarity scores between adjacent responses of the conversation.

6.3 Information Gain

Information gain reflects the number of unique words that are introduced into the conversation from both the system and the user. We believe that the more information the conversation has, the better the conversational quality is. This metric is calculated automatically by counting the number of unique words after the utterance is tokenized.

7 Results and Analysis

We evaluate the three policies with respect to three evaluation metrics: *turn-level appropriateness*, *conversational depth* and *information gain*. We show the results in the simulated setting in Table 6 and the real-world setting in Table 7. In the simulated setting, users are simulated using a

Policy	Appropriateness	Conversational depth	Info gain
Random Selection	62%	32%	50.2
Locally Greedy	72%	34%	62.4
Reinforcement Learning	82%	45%	68.2

Table 6: Performance of different policies in the simulated setting

Policy	App	Inter	Inapp	Conversational depth	Info gain
Random Selection	30%	36%	32%	30%	56.3
Locally Greedy	30%	42%	27%	52%	71.7
Reinforcement Learning	34%	43%	23%	58%	73.2

Table 7: Performance of different policies in the real-world setting.

chatbot, A.L.I.C.E.. We show an example simulated conversion in Table 4. In the real-world setting, the users are people recruited on Amazon Mechanical Turk. We collected 50 conversations for each policy. We compute *turn-level appropriateness* and *conversational depth* using automatic predictors in the simulated setting and use manual annotations in the real-world setting.

The policy learned via reinforcement learning outperforms the other two policies in all three metrics with statistical significance ($p < 0.05$)in both the simulated setting and the real-world setting. The percentage of inappropriate turns decreases when the policy considers context in selecting strategies. However, the percentage of appropriate utterances is not as high as we hoped. This is due to the fact that in some situations, no generic strategy is appropriate. For example, none of the strategies can produce an appropriate response for a content-specific question, such as "What is your favorite part of the movie?" However, the *end* strategy can produce a response, such as: "Sorry, I don't know, tell me something you are interested." This strategy is considered "Interpretable" which in turn saves the system from a breakdown. The goal of designing strategies and policies is to avoid system breakdowns, so using the *end* strategy is a good choice in such a situation. These generic strategies are designed to avoid system breakdowns, so some times they are not "Appropriate", but only "Interpretable".

Both the reinforcement learning policy and the locally greedy policy outperform the random selection policy with a huge margin in conversational depth. The reason is that they take context into consideration in selecting strategies, while the random selection policy uses the *switch* strategy randomly without considering the context. As a result, it cannot keep the user on the same topic for long. However, the reinforcement learning policy only outperforms the locally greedy policy with a small margin. Because there are cases when the user has very little interest in a topic, the reinforcement learning policy will switch the topic to satisfy the *turn-level appropriateness* metric, while the locally greedy policy seldom selects the *switch* strategy according to the learned statistics.

The reinforcement learning policy has the best performance in terms of information gain. We believe the improvement mostly comes from using the *more* strategy in the right context. The *more* strategy elicits more information from the user compared to the other general diversion strategies.

In Table 4, we can see that the simulated user is not as coherent as a human user. In addition, the simulated user is less expressive than a real user, so the depth of the conversation is generally lower in the simulated setting than in the real-world setting.

8 Conclusion

We designed a set of generic conversational strategies, such as switching topics and grounding on named-entities, to handle possible system breakdowns to power non-task-oriented systems. We also learned a policy that considers both the local and the global context of the conversation for strategy selection using reinforcement learning methods. The policy learned by reinforcement learning outperforms the locally greedy policy and the random selection policy with respect to three evaluation metrics: turn-level appropriateness, conversational depth and information gain.

References

Rafael E Banchs and Haizhou Li. 2012. Iris: a chat-oriented dialogue system based on the vector space model. In *Proceedings of the ACL 2012 System Demonstrations*, pages 37–42. Association for Computational Linguistics.

Alan W Biermann and Philip M Long. 1996. The composition of messages in speech-graphics interactive systems. In *Proceedings of the 1996 International Symposium on Spoken Dialogue*, pages 97–100.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

Min Chi, Kurt VanLehn, Diane Litman, and Pamela Jordan. 2011. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *User Modeling and User-Adapted Interaction*, 21(1-2):137–180.

Kallirroi Georgila and David R Traum. 2011. Reinforcement learning of argumentation dialogue policies in negotiation. In *INTERSPEECH*, pages 2073–2076.

Ryuichiro Higashinaka, Kenji Imamura, Toyomi Meguro, Chiaki Miyazaki, Nozomi Kobayashi, Hiroaki Sugiyama, Toru Hirano, Toshiro Makino, and Yoshihiro Matsuo. 2014. Towards an open-domain conversational system fully based on natural language processing. In *COLING*, pages 928–939.

Clayton J Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International AAAI Conference on Weblogs and Social Media*.

Filip Jurčíček, Blaise Thomson, and Steve Young. 2012. Reinforcement learning for parameter estimation in statistical spoken dialogue systems. *Computer Speech & Language*, 26(3):168–192.

Sungjin Lee and Maxine Eskenazi. 2012. Pomdp-based let's go system for spoken dialog challenge. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 61–66. IEEE.

Esther Levin, Roberto Pieraccini, and Wieland Eckert. 1997. Learning dialogue strategies within the markov decision process framework. In *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 72–79. IEEE.

Jan Niehues Maria Schmidt and Alex Waibel. 2015. Towards an open-domain social dialog system. In *Proceedings of the 6th International Workshop Series on Spoken Dialog Systems*, pages 124–129.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Teruhisa Misu, Komei Sugiura, Kiyonori Ohtake, Chiori Hori, Hideki Kashioka, Hisashi Kawai, and Satoshi Nakamura. 2010. Modeling spoken decision making dialogue and optimization of its dialogue strategy. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 221–224. Association for Computational Linguistics.

Teruhisa Misu, Kallirroi Georgila, Anton Leuski, and David Traum. 2012. Reinforcement learning of question-answering dialogue policies for virtual museum guides. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 84–93. Association for Computational Linguistics.

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*, pages 583–593. Association for Computational Linguistics.

Tomohide Shibata, Yusuke Egashira, and Sadao Kurohashi. 2014. Chat-like conversational system based on selection of reply generating module with reinforcement learning. In *Proceedings of the 5th International Workshop Series on Spoken Dialog Systems*, pages 124–129.

Satinder P Singh, Michael J Kearns, Diane J Litman, and Marilyn A Walker. 1999. Reinforcement learning for spoken dialogue systems. In *Nips*, pages 956–962.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. ICML Deep Learning Workshop 2015.

Marilyn A Walker, Jeanne C Fromer, and Shrikanth Narayanan. 1998. Learning optimal dialogue strategies: A case study of a spoken dialogue agent for email. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*, pages 1345–1351. Association for Computational Linguistics.

Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

Zhou Yu, Ziyu Xu, Alan Black, and Alexander Rudnicky. 2016. Chatbot evaluation and database expansion via crowdsourcing. In *Proceedings of the chatbot workshop of LREC*.