Building Effective Goal-Oriented Dialogue Agents

Derek Chen
Stanford University
Department of Computer Science
derekchen14@cs.stanford.edu

Abstract

Recent progress has seen an explosion in dialog systems, including voice-activated bots, text-based chatbots, and email-based bots for scheduling meetings. However real life interaction with these so-called intelligent agents consistently fall short of expectations. Consequently, this project deploys neural-based bots and rule-based bots across a wide variety of tasks in order to analyze where the behavior between the systems differ. We find that even without excessive handcrafted logic, rules-based methods can manage to perform on par with neural-based methods. Based on these results, we pinpoint strategic areas of improvement for neural-based goal-oriented dialog agents.

1. Introduction

Goal-oriented dialog systems cover a wide range of applications to boost productivity including IT tech support, healthcare consulting and calendar management. While a tremendous amount of progress has recently been made in applying deep learning techniques to natural language activities, more often than not, traditional information retrieval systems achieve superior performance over neural-based models [6]. RNN-based models often output safe generic response as a result of trying to mimic human language rather than an actual understanding of the user request [8]. Thus, industry practitioners predominantly lean on conventional machine learning methods when deploying dialog agents in production, relegating deep learning methods to academia [17].

If the gap in performance were bridged, numerous applications and services, such as call centers, could be automated or performed at drastically reduced cost. Consequently, the aim of this project is to analyze what causes goal-oriented dialog systems to outperform one another on a variety of tasks and to develop theories on why human-level performance remains elusive. More specifically, we will dive into understanding when neural-based systems break down and when they succeed over rule-based systems.

The input into our algorithms are dialogues between two partners, where we will generally refer to humans speakers as “users” and bots as “agents”. Each dialog consists of roughly 7-10 utterances where each utterance is a series of tokens (see 2). More concretely, for the neural models, an encoder will take user queries in the form of word embeddings as input and a decoder will produce vocabulary indexes as output, which are joined together to form agent responses. For the rule-based models, the raw utterance will serve as the input and a slot-filled template will be returned as output.

2. Approach

In order to analyze dialog agent effectiveness, we design a series of chat-related tasks and train RNN models to complete those tasks. Next, we examine the shortcomings of those models to build information retrieval (IR) systems to complete the same task. Critically, the IR system does not contain copious amounts of task-specific rules, and in fact attempts to operate under a generalizable discourse framework. Then, if the IR system is able to perform well, we have reason to believe that the factors that cause it to do better are highly likely to be the differences we have engineered. We hope to find key determinants of success where researchers and industry professionals can focus their efforts over less fruitful endeavors.
2.1. Neural-based Methods

The neural-based models are all variations of Sequence-to-Sequence models, which include an encoder and a decoder that are both composed of RNN units [16]. One of the four tasks use GRU units [2] whereas the remaining three use LSTM units [4] where each loop through the LSTM acts as a layer of a neural network. For any given task, the encoder will take in the user input to generate the hidden state which stores the implicit user intent. This differs from traditional goal-oriented dialog state-tracking which explicitly store user beliefs [18]. With the last hidden state of the encoder as its initial input, the decoder then outputs tokens at each time step in order to generate an agent response. All models also use an attention mechanism [11] in order to gain a better understanding of the spoken context before generating an output.

2.2. Rule-based Methods

A crucial element to this project relies on finding the right balance of features for the rule-based systems. On the one hand, if we do not add enough rules to the model, it will fail to complete the given task or appear too robotic. On the other hand, if we add too many rules, we will fail to prove our point that certain strategic features are sufficient for outperforming neural-based systems. To address the latter issue, we introduce a standard framework all rule-based models must adhere to, effectively acting as a form of regularization to limit the hypothesis space.

Concretely, all the rule-based systems follow a pattern of three phases when conversing with the user. To start, a lexer tokenizes the raw utterance and identifies the logical forms most pertinent to the conversation. For example, in a conversation about movies, the system might try to recognize movie titles or actors. In the second phase, a parser analyzes the dialog tokens to uncover the users intent. For example, a user asking about our opinion of a certain movie would get tagged as an "Inquire" intent. Finally, a manager selects from a set of templates based on the previously identified intent and fills in the slots as needed to generate an output for the user. Continuing with the last example, given the user inquiry, the manager would likely retrieve an "Inform" template. In short, all the rule-based bots must follow a framework of (1) tokenization, (2) parsing and (3) generation to produce conversation.

3. Related Work

The success of neural-based methods for goal-oriented dialog can be evidenced in [1] where they use a combination of an encoder-decoder along with a memory network to recommend restaurants to users. However even in this paper, the per dialog accuracy for most tasks remains in the single digits (out of 100%). On a more personal level, this project is inspired by the author’s own frustrations in the past when attempting to reproduce similar research [13].

Past work on the difficulty of training RNNs have discussed solutions around the exploding gradient and vanishing gradient problems [15]. More recently, research has been conducted on finding methods for overcoming the difficulties of neural text generation [19]. The present work differs from those efforts since our goal is to find high-level, theoretical areas for improvement rather than specific tips or tricks.

Diving into more specifics, the setup for two of the four tasks are taken directly from [3] and [7], namely [Mutual Friends] and [Deal or No Deal]. Therefore, their models and results are taken at face value as points of comparison. Additionally, reproducing the environment for [Deal or No Deal] was done with the support of the ParlAI environment generously open sourced by Facebook.

When designing the rule-based models, we draw inspiration from [5] whose authors find that interpretable language does not emerge naturally when communicating between two neural-based agents, and needs to be coaxed out by adding additional constraints to the problem. This highlights two insights. First, bots are trained to accomplish tasks, and not to understand the nuances of user intents or human language. Secondly, the way around this handicap is not to make the task easier, but counter-intuitively to make the task harder by restricting dialog options. Separately, the work of [10] introduces the idea of using such context when designing dialogue systems.
<table>
<thead>
<tr>
<th>Intents</th>
<th>Mutual Friends</th>
<th>Deal or No Deal</th>
<th>Craigslist Negot’n</th>
<th>Open Movies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greet</td>
<td>15.9%</td>
<td>0.3%</td>
<td>10.7%</td>
<td>4.5%</td>
</tr>
<tr>
<td>Inquire</td>
<td>15.8%</td>
<td>1.8%</td>
<td>13.3%</td>
<td>27.5%</td>
</tr>
<tr>
<td>Disagree</td>
<td>10.7%</td>
<td>0.2%</td>
<td>21.2%</td>
<td>5.7%</td>
</tr>
<tr>
<td>Agree</td>
<td>16.8%</td>
<td>27.0%</td>
<td>12.8%</td>
<td>11.0%</td>
</tr>
<tr>
<td>Inform</td>
<td>9.3%</td>
<td>62.0%</td>
<td>17.9%</td>
<td>10.7%</td>
</tr>
<tr>
<td>Unknown</td>
<td>31.6%</td>
<td>8.6%</td>
<td>24.1%</td>
<td>40.6%</td>
</tr>
</tbody>
</table>

Table 1. Identified user intents across various tasks, with remaining unmatched intents labeled as Unknown.

4. Datasets

The vast majority of training dialogues were collected by setting up a simple web app on Amazon Mechanical Turk. The user interface includes a description of the task along with a chatbox for discussion. A survey is also introduced at the end to track qualitative metrics such as the human-likeness of the agent or user. In addition to Mechanical Turk, some tasks required merging data from a variety of databases, scraping websites, and manual entry. For example, the knowledge base on popular movies was an aggregation of data from IMDB, Kaggle, Rotten Tomatoes, and Amazon movie reviews.

5. Experiments

All four experiments involve iterating between development of a neural-based model and a rule-based model. During initial stages, models are evaluated on their ability to return reasonable responses. In later passes, models are further evaluated on their ability to complete the given task and their degree of success.

5.1. Mutual Friends

In the first task, both parties have access to a separate, private knowledge base (KB) of friends, along with those friend’s attributes. For each scenario there is exactly one shared friend that occurs in both KBs, and the goal of the agents is to identify that common friend through talking to one another. In this instance, fluency refers to how grammatical and well-formed the statements are being made by the agent. Correctness refers to the ability of the agent to correctly report the data in their own KB, which is non-trivial because the KB may contain a large number of entries.

Finally both bots were roughly equal in human-likeness, although noticeably lower than actual humans. Looking at the success rate we can see that humans are quite good at this task, but that both neural and rule-based bots are not far behind. In fact, the success of the rule-based agent was slightly higher than that of the neural-based agent.

5.2. Deal or No Deal

For the second task, we turn our attention to a task devised by Lewis et al. which involves two partners deciding on how to split a group of shared items. Specifically, both players see $X$ number of books, $Y$ number of hats and $Z$ number of balls, with each item worth a certain number of points and a goal of getting as many points as possible. The twist comes in that the point values of each item is different for the two players and each player doesn’t know the values assigned to the other player. Negotiations that end in no deal lead to no reward for both users.

In the table, the Success Rate refers to the percent of dialogues that ended with a deal. Average self points refers to how many points the agent was able to achieve and conversely, average partner points refers to how many points the user was able to achieve.

During initial trials, the rule-based bot performed at a subpar rate, but the key insight we developed was recognizing that while accepting any offer would lead to fewer points, being more agreeable would lead to fewer failed negotiations. Thus, the rules were tuned so a balance could be struck between being more aggressive in the beginning and being more accepting towards the end. Critically, the system would never walk away from a deal with nothing, leading to more deals overall and better bargaining position. Unlike the
neural-based model which was always stuck mimicking optimal human behavior [3], the rules-based system could be more flexible. As a result, our system is able to hold its own against both RL-systems and humans in negotiation 3.

5.3. Craigslist Negotiation

Given the previous results, we now move on to a problem with wider scope, where we would expect the performance of the neural-based model to win out given the increased complexity. The third task emulates the experience of negotiating on Craigslist. The items being negotiated can range from small-item purchases such as outdated appliances to bigger purchases like furniture, all the way to renting apartments. Furthermore, agents and users can take on either the role of the buyer or seller with equal probability. When the dialog agent acts as a seller, the task is to get as high a sale price as possible. Alternatively, a buyer’s task is to get as low a buying price as possible.

The new metric being reported is the normalized average price, where a higher number is better and the expected outcome is defined to be 1.0. To see how we arrived at this score, first recall that item prices range from roughly $25 to $2500, which represents two orders of magnitude in price, so to deal with this issue all sale amounts were normalized to 1. Additionally, buyers are actually incentivized to find lower prices, so their scores were reversed. Finally, we average scores to make them comparable across models.

Analyzing the errors, we found the biggest gripe users had when chatting with the bot occurred when the system ignored the user’s suggested price or maintained an incorrect belief of the price. For example, a user might sell a TV for $500, which the bot would reject, but then inexplicably suggest a counter-offer of $600. This type of mistake makes sense because the neural based model does not have an explicit representation of the dollar amount, and instead stores an continuous-valued embedding holding onto a vague notion of “price”.

Armed with this insight, we designed our IR model to focus very heavily on understanding the user’s intended price. In fact, we double down on this idea such that the vast majority of rules ignore all other forms of user utterances and instead exclusively pay attention to price. As it turns out, most aspects of a user’s proposal can be safely ignored, including the type of item being sold or bargaining strategy, as long as the correct relative price is properly maintained. In doing so, the rule-based models scored as well as or better than the neural-based model 4 on average. Whereas neural-based models try to learn everything, rule-based models can be tuned to focus only on what matters.

5.4. Open Movies Discussion

Based on the success of the previous task, we decided to push the limits and moved onto open-domain chat. This time around users were allowed to discuss anything within the realm of movies, with no explicit

Table 2. Statistics on training data collected from Human-Human dialogues on Mechanical Turk

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Mutual Friends</th>
<th>Deal or No Deal</th>
<th>Craigslist Negotiation</th>
<th>Open Movies Discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Dialogues</td>
<td>9,041</td>
<td>12,234</td>
<td>6,682</td>
<td>1,680</td>
</tr>
<tr>
<td>Avg. Number of Utterances</td>
<td>11.41</td>
<td>4.98</td>
<td>7.39</td>
<td>10.58</td>
</tr>
<tr>
<td>Utterance Length (tokens)</td>
<td>5.08</td>
<td>8.85</td>
<td>13.22</td>
<td>10.89</td>
</tr>
<tr>
<td>Vocabulary Size</td>
<td>5325</td>
<td>2745</td>
<td>27873</td>
<td>9000</td>
</tr>
</tbody>
</table>
Table 5. Selected metrics for Open Movies Discussion.

<table>
<thead>
<tr>
<th>Interestigness</th>
<th>Humanlikeness</th>
<th>Error Margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural</td>
<td>3.5</td>
<td>3.5</td>
</tr>
<tr>
<td>Rule-based</td>
<td>2.7</td>
<td>3.4</td>
</tr>
</tbody>
</table>

Figure 5. Selected metrics for Open Movies Discussion.

...goal, and thus no associated success rate. As expected, the ratio of utterances with an unknown user intent was higher for [Open Movies] than for any other task.

Drawing from past experience, we kickstart the rule-based bot by limiting its scope to focusing on just movie titles and genres as key entities, excluding other relevant facts such as directors, year released, or writers. Given the sheer volume of movies ever produced, there is no reasonable method for having every film in the model’s database. Therefore, one key idea the agent used to keep the problem tractable was to be proactive in suggesting certain movies to the user. For example, the agent might guide the user to talk about favorite movies, and mention that its favorite movie is “The Lion King”. Importantly, the agent would mention “The Lion King” because it knows its KB has been populated with information about that movie.

In the end, the rule-based bot fell short of the neural-based model in terms of being neither more interesting nor more humanlike. Although it is noteworthy that even in a realm with such a broad scope, the rule-based bot still performed competitively.

6. Discussion

Recall that the goal of this project is not to build the best dialog agents possible, but rather to examine the factors that make dialog agents effective. Consequently, the results should not be judged by how high of a score any given model was able to achieve, but instead should be measured by the meaningfulness of the comparisons found between the different types of bots. Furthermore, comparisons can be considered meaningful if they led to useful insights which are then used to design the rule-based agents. To this end, the results show rule-based systems fitting within a standard framework is sufficient for performing well across a variety of dialog tasks, so we believe the experiments were a success.

In summary, the strength of the rule-based bot was accomplished with just a few strategic insights: (1) Most discussions can be understood by tracking a limited number of key entities, so if the right items have been identified, then the agent is able to ignore most other items. (2) Difficulty of open domain discussion can be mitigated by driving conversations into “safe zones” where the agent has better understanding. (3) Context is critical to conversation, but can be tracked with very small number of states. The upshot is that context and key entities are important for generating meaningful dialog, and thus improved neural based bots should work to be incorporate this knowledge when making decisions.

In hindsight, the conclusion of incorporating context seems fairly obvious, but we believe this is only the case because humans naturally infer context in their conversations, and we forget that bots only optimize for exactly what we tell it to learn. As a result, perhaps a bigger takeaway is improved models should move beyond following orders to instead make an effort at understanding what matters to their partner. Coincidentally, this mindset mirrors how negotiation is usually taught in business schools.

7. Future Work

Given the analysis, a clear next step is designing a method for neural systems to directly train on user intents and past context. Historically, keeping track of explicit user beliefs meant modular training, but we might be able to maintain end-to-end training these days using reinforcement learning. Concretely, the value function should return high rewards for correct interpretations of the user intent. At this point, the astute reader might notice that Facebook’s agent also included an RL component, however their model optimized for maximizing points during negotiation. In contrast, our experiments suggest improved results may come from optimizing for user understanding.

Another assumption to consider overturning is the idea that humans make the best agents. Specifically, all the tasks implicitly placed more value in agents that behave in a human like manner. However, what if real-life users don’t mind dealing with a bot as long as the agent effectively solves their problems. Knowing how important humanlikeness is helps prioritize how bots are built. Overall, there are many theoretical and practical issues left to explore in the aim of building effective goal-oriented dialog agents.
8. Contributions

Derek Chen (SUNetID: dchen14) wrote the entirety of this paper and large portions of the code required to run experiments. This work is done in collaboration with He He from the Stanford NLP Lab under the guidance of Percy Liang. With that said, He is not enrolled in CS229.

References


