Automatic Response Generation for Conversational e-Commerce Agents
Snehasish Mukherjee (mukherji@stanford.edu)

OBJECTIVE
A response is said to be entertaining when it follows the users way of speaking. Entertaining responses are desirable but many conversational e-commerce systems use hard-coded templates to generate responses and hence cannot entertain.

RL APPROACH
We model the problem as a game play or a control problem where the state at any given point is defined by

\[ S = w_1, w_2, \ldots, w_n \]  

where \( w_i \) s are the words of the response at that point in time. We define the following 2 types of operators on the state: \( P(S) \) - generates a new permutation of \( S \) and \( T(S, a) \) - transforms \( w_i \) including deleting it, depending on the second parameter, \( a \). We use Monte Carlo Tree Search to stochastically explore the search space. At any given time, the search state moves to a new state for which the following quantity is highest:

\[ \text{UCB1} = v_i + C \sqrt{\log N / n_i} \]

For evaluating the final generated sentence we use a multi-objective function that rewards similarity to the users original utterance while penalizing parts of speech that are indicative of a question being asked. The reward \( R \) for a given state \( s \) is given as follows

\[ R(s) = t(s) * (c * k(s) + (1 - c) * l(s) + b(s, ref)) \]

where \( t(s) \) is the output from a Naive Baye’s classifier that determines the probability of \( s \) being a valid sentence, \( k(s) \) and \( l(s) \) are smoothed probabilities from language models and \( b(s, ref) \) is the BLEU score.

RESULTS

<table>
<thead>
<tr>
<th>Length</th>
<th>Bandit</th>
<th>Score</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;= 5</td>
<td>UCT</td>
<td>0.7129</td>
<td>80.0%</td>
</tr>
<tr>
<td></td>
<td>PUCT</td>
<td>0.6990</td>
<td>70.0%</td>
</tr>
<tr>
<td></td>
<td>Random</td>
<td>0.4127</td>
<td>50.0%</td>
</tr>
<tr>
<td>5 – 10</td>
<td>UCT</td>
<td>0.6274</td>
<td>55.0%</td>
</tr>
<tr>
<td></td>
<td>PUCT</td>
<td>0.6035</td>
<td>60.0%</td>
</tr>
<tr>
<td></td>
<td>Random</td>
<td>0.3255</td>
<td>10.0%</td>
</tr>
<tr>
<td>&gt;= 11</td>
<td>UCT</td>
<td>0.5535</td>
<td>40.0%</td>
</tr>
<tr>
<td></td>
<td>PUCT</td>
<td>0.5932</td>
<td>40.0%</td>
</tr>
<tr>
<td></td>
<td>Random</td>
<td>0.3881</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

Table 1: Performance for different Bandit strategies

DISCUSSIONS

• Though some results were encouraging, the reward function needs more tuning to correlate more strongly to human evaluation.
• There is no standard implementation of Kneser-Ney smoothing for trigrams. Current NLTK implementation is buggy.
• Though it is normally ill-advised to use BLEU score at sentence level, we managed to use it to our advantage with appropriate smoothing and defensive coding.
• Naive Baye’s worked surprisingly well for valid vs invalid sentence classification, when the invalid sentences were generated by sampling randomly from the corpus.

FUTURE WORK

This is an area of active research at WalmartLabs and efforts to improve and develop this approach will continue beyond the end of this course. Following will be the main areas of focus

• In addition to deterministic set of transition rules and unsupervised NLG, we will also explore if we can use a stochastic action space and learn the best moves in a supervised setting.
• The reward function can be parameterized, with different weights for each of the component objectives, which can then be learned from a held out data set.
• We will explore Expert Iteration System that generalizes the learning from the state transitions, which can then be used as a feedback to MCTS at the simulation step.

REFERENCES