Learning suitable and well-performing dialogue behaviour in statistical spoken dialogue systems has been in the focus of research for many years. While most work which is based on reinforcement learning employs an objective measure like task success for modelling the reward signal, we propose to use a reward based on user satisfaction. We will show in simulated experiments that a live user satisfaction estimation model may be applied resulting in higher estimated satisfaction whilst achieving similar success rates. Moreover, we will show that one satisfaction estimation model which has been trained on one domain may be applied in many other domains which cover a similar task. We will verify our findings by employing the model to one of the domains for learning a policy from real users and compare its performance to policies using the user satisfaction and task success acquired directly from the users as reward.

Index Terms: spoken dialogue systems, statistical dialogue management, interaction quality, reinforcement learning

1. Introduction

For modelling the decision-making component of a spoken dialogue system (SDS), the dialogue policy, different approaches exist. A very prominent one is to model the problem as a (partially observable) Markov decision process (POMDP) using reinforcement learning (RL) to learn the optimal system behaviour. In RL, the policy \( \pi \) is trained to make decisions so that a potentially delayed objective (the reward function) is maximised. For information-seeking dialogues, most existing work has been used in prior work precisely because it does not have any information available about the goal of the dialogue. The model has been trained on manually annotated dialogue turns of a bus information system achieving an accuracy of 0.89\(^2\). The proposed RL framework is shown in Figure 1. It has previously been applied for in-domain experiments and simulated evaluation [7]. In this paper, we will complete the work by using a modified reward function to show its domain-independence and the resulting high potential to be applicable for learning in unseen domains. Moreover, the estimator is used in an experiment where the policy is learned through interaction with real humans.

Most of previous work focuses on employing task success as the main reward signal [9, 10, 11, 12, 13, 14, 15, 16]. However, task success is usually only computable for predefined tasks e.g., through interactions with simulated or recruited users, where the underlying goal is known in advance. To overcome this, the required information can be requested directly from users at the end of each dialogue [17]. However, this can be intrusive, and users may not always cooperate.

An alternative is to use a task success estimator [18, 15, 16]. With the right choice of features, these can also be applied to new and unseen domains [19]. However, these models still attempt to estimate completion of the underlying task, whereas our model evaluates the overall user experience.

In this paper, we show that an interaction quality reward estimator trained on dialogues from a bus information system will result in well-performing dialogues both in terms of success rate and user satisfaction on five other domains, while only using interaction-related, domain-independent information, i.e., not knowing anything about the task of the domain.

Others have previously introduced user satisfaction into

\(^1\)The relation of US and IQ has been closely investigated in [2, 8].
\(^2\)taking into account neighbouring values, cf. Sec. 2
the reward [20, 21, 22, 23] by using the PARADISE framework [24]. However, to derive user ratings within that framework, users have to answer a questionnaire which is usually not feasible in real world settings. To overcome this, PARADISE has been used in conjunction with expert judges instead [25, 26] to enable unintrusive acquisition of dialogues. However, the problem of mapping the results of the questionnaire to a scalar reward value still exists.

Furthermore, PARADISE assumes a linear dependency between measurable parameters and user satisfaction whereas a non-linear dependency might be more appropriate [27]. Therefore, we use interaction quality [2] in this work because it uses scalar values applied by experts and assumes a non-linear dependency between measurable parameters and the target value.

The remainder of the paper is organized as follows: in Section 2, the interaction quality reward estimation module is presented in detail. Section 3 contains the simulated experiments on several domains as well as an experiment with paid subjects. The findings are discussed in Section 4 and conclusions are drawn in Section 5.

2. Interaction Quality Reward Estimation

In this work, we propose to use interaction quality (IQ) [2] as a reward estimator for learning information-seeking dialogue policies. IQ represents a less subjective variant of user satisfaction: instead of being acquired from users directly, experts annotate pre-recorded dialogues to avoid the large variance that is often encountered when users rate their dialogues directly [2].

IQ is defined on a five-point scale from five (satisfied) down to one (extremely unsatisfied). To derive a reward from this value, the equation

$$R_{IQ} = T \cdot (-1) + (iq - 1) \cdot 5$$  (1)

is used where $R_{IQ}$ describes the final reward. It is applied to the final turn of the dialogue of length $T$ with a final IQ value of $iq$. Thus, a per-turn penalty of $-1$ is added to the dialogue outcome. This results in a reward range of 19 down to $-T$ which is consistent with related work [9, 19, 16, e.g.] in which binary task success (TS) was used to define the reward as:

$$R_{TS} = T \cdot (-1) + \sum_{TS} \cdot 20$$  (2)

where $\sum_{TS} = 1$ only if the dialogue was successful, $\sum_{TS} = 0$ otherwise. $R_{TS}$ will be used as a baseline.

The problem of estimating IQ is cast as a classification problem where the target classes are the distinct IQ values. The input consists of domain-independent variables called interaction parameters. These parameters incorporate information from the automatic speech recognition (ASR) output and the preceding system action. Based on this information, which is available at every turn, temporal features are computed taking sums, means or counts from the turn-based information for a window of the last 3 system-user-exchanges and the complete dialogue (see Fig. 2). This results in a feature set of 16 parameters as shown in Table 1.

As training data, the LEGO corpus [28] is used which consists of 200 dialogues (4,885 turns) from the Let’s Go bus information system [29]. There, users with real needs are able to call the system to get information about the bus schedule. Each turn of these 200 dialogues has been annotated with IQ (representing the quality of the dialogue up to the current turn) by three experts. The final IQ label has been assigned using the median of the three individual labels.

The estimation model was trained using a support vector machine [30, 31] achieving an unweighted average recall (UAR) of 0.55 with 10-fold cross-validation. However, missing the correct estimated IQ value by only one has little impact for modelling the reward, and if neighbouring values are taken into account, the model achieves an accuracy of 0.89.

As a comparison, previous work has used the LEGO corpus with a full IQ feature set (which includes additional partly domain-related information) and this achieves a UAR of 0.55 using ordinal regression [32], 0.53 using a two-level SVM approach [33], and 0.51 using a hybrid-HMM [34]. Human performance on the same task is 0.69 UAR [2].

3. Experiments and Results

The proposed IQ reward estimation framework (Fig. 1) was evaluated on several domains within a simulated environment. Furthermore, the simulated results were validated by applying the framework to one of the domains and learning the policy directly through interaction with real humans.

3.1. Experimental Setup

To train and evaluate the proposed framework, a policy model based on the GP-SARSA algorithm [9] is used. This is a value-based method that uses a Gaussian process to approximate the state-value function. As it takes into account the uncertainty of the approximation, it is very sample efficient and may even be used to learn a policy directly through real human interaction [17]. The decisions of the policy are based on a summary

---

---

---
policies on five different domains was evaluated: Cambridge For the simulation experiments, the performance of the trained
3.2. Domain-independent Learning from Simulation on the estimated IQ at the end of each dialogue.

The exact number of system actions varies for the domains and
request are based on general intents like looking for?”) which has been used as the baseline for
A second baseline was also included: directly acquiring a user satisfaction (US) rating from the users after each dialogue. For this, the second question posed was: “How satisfied are you with the interaction?” The users were able to respond on a six-point scale: 6=very satisfied, 5=satisfied, 4=generally ok, 3=unsatisfied, 2=very unsatisfied or 1=extremely unsatisfied. This rating was converted to a reward in correspondence with \( R_{US} \): 

\[
R_{US} = T \cdot (-1) + (US - 1) \cdot 5 .
\]

Hence, each dialogue was also evaluated using the average user satisfaction (AUS).

Two policies were trained for each reward function. The learning curves show moving TSR, moving AIQ and moving AUS and are presented in Figure 3. Each value in the graphs is

Table 2: Statistics of the domain the IQ reward estimator is trained on (LetsGo) and the domains it is applied to.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Code</th>
<th># constraints</th>
<th># DB items</th>
</tr>
</thead>
<tbody>
<tr>
<td>LetsGo</td>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CamRestaurants</td>
<td>CR</td>
<td>3</td>
<td>110</td>
</tr>
<tr>
<td>CamHotels</td>
<td>CH</td>
<td>5</td>
<td>33</td>
</tr>
<tr>
<td>SFRestaurants</td>
<td>SR</td>
<td>6</td>
<td>271</td>
</tr>
<tr>
<td>SFHotels</td>
<td>SH</td>
<td>6</td>
<td>182</td>
</tr>
<tr>
<td>Laptops</td>
<td>L</td>
<td>6</td>
<td>126</td>
</tr>
</tbody>
</table>

Table 3: Results of the simulated experiments for all domains showing task success rate (TSR), average interaction quality (AIQ), and average dialogue length (ADL) in number of turns. Each value is computed after 100 evaluation / 1,000 training dialogues averaged over three trials. * marks statistically significant difference between \( R_{TS} \) and \( R_{IQ} \) (\( p < 0.05 \), T-test).

<table>
<thead>
<tr>
<th>Domain</th>
<th>SER</th>
<th>TSR</th>
<th>AIQ</th>
<th>ADL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0%</td>
<td>0.98</td>
<td>0.98</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15%</td>
<td>0.99</td>
<td>0.96</td>
<td>3.02</td>
</tr>
<tr>
<td></td>
<td>30%</td>
<td>0.97</td>
<td>0.96</td>
<td>3.03</td>
</tr>
<tr>
<td>CR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0%</td>
<td>0.99</td>
<td>0.93</td>
<td>2.58</td>
</tr>
<tr>
<td></td>
<td>15%</td>
<td>0.93</td>
<td>0.94</td>
<td>3.11</td>
</tr>
<tr>
<td></td>
<td>30%</td>
<td>0.84</td>
<td>0.84</td>
<td>2.61</td>
</tr>
<tr>
<td>CH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0%</td>
<td>0.99</td>
<td>0.93</td>
<td>2.58</td>
</tr>
<tr>
<td></td>
<td>15%</td>
<td>0.93</td>
<td>0.94</td>
<td>3.11</td>
</tr>
<tr>
<td></td>
<td>30%</td>
<td>0.84</td>
<td>0.84</td>
<td>2.61</td>
</tr>
<tr>
<td>SR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0%</td>
<td>0.99</td>
<td>0.93</td>
<td>2.58</td>
</tr>
<tr>
<td></td>
<td>15%</td>
<td>0.93</td>
<td>0.94</td>
<td>3.11</td>
</tr>
<tr>
<td></td>
<td>30%</td>
<td>0.84</td>
<td>0.84</td>
<td>2.61</td>
</tr>
<tr>
<td>SH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0%</td>
<td>0.99</td>
<td>0.93</td>
<td>2.58</td>
</tr>
<tr>
<td></td>
<td>15%</td>
<td>0.93</td>
<td>0.94</td>
<td>3.11</td>
</tr>
<tr>
<td></td>
<td>30%</td>
<td>0.84</td>
<td>0.84</td>
<td>2.61</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0%</td>
<td>0.99</td>
<td>0.93</td>
<td>2.58</td>
</tr>
<tr>
<td></td>
<td>15%</td>
<td>0.93</td>
<td>0.94</td>
<td>3.11</td>
</tr>
<tr>
<td></td>
<td>30%</td>
<td>0.84</td>
<td>0.84</td>
<td>2.61</td>
</tr>
<tr>
<td>TV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0%</td>
<td>0.99</td>
<td>0.93</td>
<td>2.58</td>
</tr>
<tr>
<td></td>
<td>15%</td>
<td>0.93</td>
<td>0.94</td>
<td>3.11</td>
</tr>
<tr>
<td></td>
<td>30%</td>
<td>0.84</td>
<td>0.84</td>
<td>2.61</td>
</tr>
</tbody>
</table>

space representation of the dialogue state tracker. In this work, the focus tracker [35]—an effective rule-based tracker—is used. The policy may choose out of a set of summary actions which are based on general intents like request, confirm or inform. The exact number of system actions varies for the domains and ranges from 16 to 25.

The IQ reward estimator is evaluated against the baseline of using the traditional reward function based on task success (TS). While IQ needs to be estimated, TS can be computed by comparing the outcome of each dialogue with the pre-defined goal. Of course, this is only possible in simulation and when evaluating with paid subjects. This goal information is not available to the IQ estimator, nor is it required.

To measure the dialogue performance, the task success rate (TSR) and the average interaction quality (AIQ) are measured: the TSR represents the ratio of dialogues for which the system satisfies, 2=very unsatisfied or 1=extremely unsatisfied. This label is noisy, only the dialogues where this subjective success label matches the objective success were used for policy training [17] (\( obj = subj \)).

In comparison to the task success estimator proposed by Vandyke et al. [19] who trained the estimator on CR and applied it to SF and SH achieving comparable results, the model proposed here does not require the maximum number of slots to be defined, i.e., the features which have been used for the user satisfaction estimator are independent of the slots.

Table 2: Statistics of the domain the IQ reward estimator is trained on (LetsGo) and the domains it is applied to.

3.3. Learning from Real Humans

For learning a policy directly from the interaction with real humans, the CR domain was used. Using the Amazon Mechanical Turk, subjects were recruited to talk to the telephone-based dialogue system. At the end of each dialogue, users were asked two questions. The first was a yes/no question targeting the dialogue success (“Have you found all the information you were looking for?”) which has been used as the baseline for \( R_{TS} \). As this label is noisy, only the dialogues where this subjective success label matches the objective success were used for policy training [17] (\( obj = subj \)).

A second baseline was also included: directly acquiring a user satisfaction (US) rating from the users after each dialogue. For this, the second question posed was: “How satisfied are you with the interaction?” The users were able to respond on a six-point scale: 6=very satisfied, 5=satisfied, 4=generally ok, 3=unsatisfied, 2=very unsatisfied or 1=extremely unsatisfied. This rating was converted to a reward in correspondence with \( R_{US} \):

\[
R_{US} = T \cdot (-1) + (US - 1) \cdot 5 .
\]

Hence, each dialogue was also evaluated using the average user satisfaction (AUS).

The results nicely show the successful evaluation of the policies using the IQ reward estimator in terms of TSR and AIQ. For the domains CR, CH, SR, and SH, the TSRs of \( R_{IQ} \) are very similar to the TSRs of \( R_{TS} \) for an SER of 0%. This slightly degrades for higher SERs. This behaviour may be attributed to the following two reasons: the more the source domain (LetsGo) of the estimator and the target domain differ, the more the results differ in terms of TSR. Furthermore, the higher the noise, the more the policy has to focus on success.

3.3. Learning from Real Humans

For learning a policy directly from the interaction with real humans, the CR domain was used. Using the Amazon Mechanical Turk, subjects were recruited to talk to the telephone-based dialogue system. At the end of each dialogue, users were asked two questions. The first was a yes/no question targeting the dialogue success (“Have you found all the information you were looking for?”) which has been used as the baseline for \( R_{TS} \). As this label is noisy, only the dialogues where this subjective success label matches the objective success were used for policy training [17] (\( obj = subj \)).

A second baseline was also included: directly acquiring a user satisfaction (US) rating from the users after each dialogue. For this, the second question posed was: “How satisfied are you with the interaction?” The users were able to respond on a six-point scale: 6=very satisfied, 5=satisfied, 4=generally ok, 3=unsatisfied, 2=very unsatisfied or 1=extremely unsatisfied. This rating was converted to a reward in correspondence with \( R_{US} \):

\[
R_{US} = T \cdot (-1) + (US - 1) \cdot 5 .
\]

Hence, each dialogue was also evaluated using the average user satisfaction (AUS).

Two policies were trained for each reward function. The learning curves show moving TSR, moving AIQ and moving AUS and are presented in Figure 3. Each value in the graphs is

5The satisfaction will be less different in noisy channels between a dialogue which is successful but has a lot of ASR non-understandings compared to a dialogue that is not successful. Thus, there might be an upper bound for the satisfaction if there is noise in the channel with the consequence that satisfaction plays a reduced role in training.
Figure 3: Moving TSR (left), moving AIQ (middle) and moving AUS (right) for using either TS, IQ, or US as reward averaged over two policies respectively, computed on windows consisting of 120 dialogues.

| Sys | Welcome to the Cambridge restaurant dialogue system. How may I help you? |
| Usr | i’m looking for european food in the expensive price range |
| Sys | Welcome to the Cambridge restaurant dialogue system. How may I help you? |
| Usr | i want to european restaurant |
| Sys | Would you like something in the cheap, moderate, or expensive price range? |
| Usr | i don’t care the area |
| Sys | Would you like something in the cheap, moderate, […]? |
| Usr | european expensive food |
| Sys | Would you like something in the cheap, moderate, […]? |
| Usr | european food |
| Sys | What part of town do you have in mind? |
| Usr | i don’t care |
| Sys | michaelhouse cafe is a nice place […] . |
| Usr | thank you goodbye |

Figure 4: Example dialogue trained using $R_{TS}$ after 387 training dialogues resulting in $iq = 1$.

Calculated using the respective previous 120 training dialogues averaged over the three trials.

The results show that $R_{IQ}$ may successfully be used for learning through interactions with real users. It achieves the same TSR compared to the baseline of $R_{TS}$ while resulting in better results for AIQ. Furthermore, the results for moving AUS indicate a slightly better user satisfaction compared to $R_{TS}$.

The second baseline of using $R_{US}$ also resulted in competitive results showing a similar TSR as $R_{IQ}$ and $R_{TS}$ while resulting in slightly better AUS. This indicates that for a problem as defined by the CR domain, using user satisfaction directly as a learning signal is also a viable option for learning policies.

Figures 4 and 5 show two successful example dialogues for the models trained with $R_{TS}$ and $R_{IQ}$, respectively. One effect of training with $R_{IQ}$ was a reduced number of system repetitions (which may be linked to the RePrompt? feature).

4. Discussion

A key aspect of this work to emphasise is that the estimator works without any knowledge about the domain. So, in contrast to task success estimators [15, 19], it does not use the dialogue state as input. Simply by using parameters encoding interaction characteristics, a dialogue policy was trained to achieve not only good US (which is optimised on) but also a good TSR.

One limitation of the proposed IQ reward estimator is that it requires manual annotation of dialogues with interaction quality labels. These labels incur a higher annotation cost than success labels. However, for this work, a total of only 200 annotated dialogue were sufficient to create a model that was able to be used on several different domains. In contrast, training a task success estimator based on recurrent neural networks typically requires 1,000 annotated dialogues [15, 19].

5. Conclusion

This work has shown that employing a user satisfaction reward estimator for learning dialogue policies without any knowledge about the domain can yield good performance in terms of both task success rate and (estimated) user satisfaction. This has been demonstrated by training the reward estimator on a bus information domain and applying it to learn dialogue policies in five different domains (Cambridge restaurants and hotels, San Francisco restaurants and hotels, Laptops) in a simulated experiment. Moreover, the estimator has successfully been applied to learning dialogue policies in the domain of finding a restaurant in Cambridge through interaction with real users.

For future work, the problem of degrading performance if the noise level increases should be tackled. One possible solution would be to have a combination of success and satisfaction as the reward. In addition, active learning will be investigated to mitigate the requirement for IQ annotated training data.

6. Acknowledgements

This research was funded by the EPSRC grant EP/M018946/1 Open Domain Statistical Spoken Dialogue Systems.
7. References

[1] J. D. Williams and S. J. Young, “Characterizing task-oriented dial-
log using a simulated asr channel,” in Proc. of the 8th Interspeech, 2004, pp. 185–188.

ity of ongoing spoken dialogue interaction by experts—and how it relat-

fical framework for a user-centered spoken dialogue manager,” in Proceed-
ings of the Paralinguistic Information and its Integrat-
ion in Spoken Dialogue Systems Workshop, R. Lópe-

spoken dialogue management,” in NAACL-HLT Workshop on Fu-

Based Human-Computer Interaction, A. I. Rudnicky, A. Raux, I. Lane, and T. Misi, Eds. Cham: Springer International Pub-


dialogue systems – experts vs. users,” in Proc. of NAACL-

dialogue manager optimization,” IEEE/ACM Transactions on Au-


logue systems,” in Proc. of the 8th Interspeech, 2007, pp. 2685–
2688.


misation of Bayesian spoken dialogue systems via human interact-


[23] ——, “Automatic learning and evaluation of user-centered objec-
tive functions for dialogue system optimisation,” in Proc. of 6th LREC, N. Calzolari, K. Choukri, and B. Maegaard, Eds. Mar-


ing for Dialogue Management,” in Proc. of the 6th Starting AI Re-


[28] A. Schmitt, S. Ultes, and W. Minker, “A parameterized and an-
notated spoken dialog corpus of the cmu let’s go bus information


[31] L. El Asri, H. Khouzaimi, R. Laroche, and O. Pietquin, “Or-
dinal regression for interaction quality prediction,” in Inter-

tion Using Error Correction,” in Proc. of the 14th SIGDial Confer-

[33] ——, “Interaction Quality Estimation in Spoken Dialogue Sys-
tems Using Hybrid-HMMs,” in Proc. of the 15th SIGDial Confer-


tem toolkit,” in Proc. of the 55th ACL. Demos. ACL, 2017, soft-

[36] J. Schatzmann and S. J. Young, “The hidden agenda user simu-