Comparing Regression Methods for Dialogue System Evaluation on a Richly Annotated Corpus

Kallirroi Georgila
Institute for Creative Technologies, University of Southern California
12015 Waterfront Drive, Los Angeles, CA 90094-2536, USA
kgeorgila@ict.usc.edu

Abstract

We compare various state-of-the-art regression methods for predicting user ratings of their interaction with a dialogue system using a richly annotated corpus. We vary the size of the training data and, in particular for kernel-based methods, we vary the type of kernel used. Furthermore, we experiment with various domain-independent features, including feature combinations that do not rely on complex annotations. We present detailed results in terms of root mean square error, and Pearson's r and Spearman's ρ correlations. Our results show that in many cases Gaussian Process Regression leads to modest but statistically significant gains compared to Support Vector Regression (a strong baseline), and that the type of kernel used matters. The gains are even larger when compared to linear regression. The larger the training data set the higher the gains but for some cases more data may result in over-fitting. Finally, some feature combinations work better than others but overall the best results are obtained when all features are used.

1 Introduction

Dialogue evaluation is an important research topic which over the years has received much attention but still remains an unsolved problem. This is because the quality of a human-machine interaction can be influenced by a large number of factors, such as the genre or domain of dialogue, the design and capabilities of the system and its components, the user expertise and expectations, etc.

In this paper we focus on task-oriented dialogue and our goal is to predict user satisfaction, i.e., user ratings after interacting with the dialogue system. For this purpose we use a richly annotated dialogue corpus with contextual information, and speech act and task labels. This corpus was derived from the original COMMUNICATOR corpus (Walker et al., 2001a) via automatic annotation (Georgila et al., 2005b, 2009). Users of the COMMUNICATOR systems try to book a flight and they may also make hotel or car-rental arrangements. An example dialogue excerpt is shown in Figure 2 in the Appendix.

The original COMMUNICATOR corpus contained speech act and task annotations for the system's side of the conversation based on the DATE scheme (Walker and Passoneau, 2001). Georgila et al. (2005b, 2009) added speech act and task annotations for the user's side of the conversation, as well as information about the dialogue context, e.g., filled slots, filled slots values, grounded slots, speech acts history, etc. The corpus consists of dialogues collected between human users and 8 dialogue systems. We extract domain-independent features from this corpus, and perform regression experiments in order to predict 5 different types of user satisfaction ratings. The corpus and the features we use are discussed in Section 3.

We explore 3 research questions: (i) Which regression method works best and does the choice of kernel matter for kernel-based regression? (ii) What is the impact of varying the training data size? (iii) Which feature combinations work best?

Our contributions are as follows: (1) We compare various state-of-the-art regression methods, in particular, linear regression, linear ridge regression, Support Vector Regression (SVR), and Gaussian Process Regression (GPR). We also vary the kernel type for GPR. To our knowledge, GPR has never been used before for dialogue system evaluation (or generally by the dialogue community) despite the fact that it is considered as the state-of-the-art for regression in other research areas. (2) We vary the size of the training data and report on its impact on performance for all regression methods. (3) We vary the feature combinations used and discuss how the choice of features affects the prediction quality of our models. Our features are domain-independent but are derived from a richly annotated corpus with dialogue context and history, and speech act and task labels. Even though

Proceedings of the 26th Workshop on the Semantics and Pragmatics of Dialogue, August, 22-24, 2022, Dublin.

the features we use are domain-independent, our experiments provide valuable insights about the benefits of different feature combinations, including features taking into account dialogue context and dialogue history, as well as features that are not dependent on complex annotations.

Our results show that in many cases GPR leads to modest but statistically significant gains compared to SVR (a strong baseline), and that the type of kernel matters. The gains are even larger when compared to linear regression. The larger the training set the higher the gains but for some cases more data may result in over-fitting. Some feature combinations work better than others but overall the best results are obtained when all features are used.

2 Related Work

Dialogue evaluation is an important area of research, and over the years there have been various surveys recording the state-of-the-art, challenges, and future directions in this research area (Hastie, 2012; Deriu et al., 2021; Mehri et al., 2022).

Prior to the recent advancement of chatbots, most research on dialogue evaluation focused on measuring the quality of human-system dialogue interaction mainly for task-oriented dialogue systems. Dialogue evaluation metrics can be subjective (e.g., user satisfaction, perceived task completion, etc.), or objective (e.g., word error rate, dialogue length, etc.). Interaction logs provide information for calculating objective measures whereas subjective assessments can be collected via surveys and questionnaires (Hone and Graham, 2000).

The most well-known framework for automating the dialogue evaluation process is PARADISE (Walker et al., 2000). PARADISE aims to optimize a desired quality such as user satisfaction by formulating it as a linear combination of various metrics, such as task success and dialogue cost (e.g., dialogue length, speech recognition errors, etc.). The contribution of each factor is determined by weights calculated via linear regression. The advantage of this method is that once a desired quality has been formulated as a realistic evaluation function, it can be optimized by controlling the factors that affect it. Thus, user satisfaction can for example be optimized by increasing task success, and minimizing dialogue length and speech recognition errors. Note however that longer dialogue lengths are not necessarily indicative of poor dialogue quality but depending on the task they may actually indicate user engagement and satisfaction (Foster et al., 2009). Indeed, PARADISE has been shown to be capable of automatically predicting dialogue quality in the travel planning domain (Wright-Hastie et al., 2002). However, it has been argued that PARADISE cannot accurately predict individual user judgements and only covers 40-50% of the variance in the data that it is trained on (Möller and Ward, 2008).

In non-task-oriented dialogue systems (e.g., chatbots) developing robust evaluation metrics can be even harder than for task-oriented dialogue. Here it is not clear what success means and task-specific objective metrics are not appropriate. Instead subjective evaluations for appropriateness of responses can be much more meaningful, which has led to the development of coding schemes for response appropriateness in such cases (Traum et al., 2004; Robinson et al., 2010).

Currently, word-overlap similarity metrics such as BLEU, METEOR, and ROUGE (originally employed in machine translation and summarization) are widely used for measuring chatbot dialogue quality. However, it has been shown that BLEU, METEOR, and ROUGE do not correlate well with human judgements of dialogue quality (Liu et al., 2016). Discriminative BLEU, a variation of BLEU where reference strings are scored for quality by human raters, was found to correlate better with human judgements than standard BLEU (Galley et al., 2015). To address the issues with BLEU, METEOR, and ROUGE, next utterance classification was introduced as a method for evaluating chatbots (Lowe et al., 2016), but the proposed metric recall@k does not take into account the fact that just because a system response is not part of a pre-defined set of appropriate responses it does not mean that it is wrong. Furthermore, topic-based metrics for chatbot evaluation (topic breadth and topic depth) were found to correlate well with human judgements (Guo et al., 2017).

There has also been work on estimating user satisfaction at the system-user exchange level rather than rating the whole dialogue (Engelbrecht et al., 2009; Higashinaka et al., 2010; Ultes and Minker, 2014; Schmitt and Ultes, 2015). Recently, new evaluation metrics have been proposed for open-domain dialogue leveraging pre-trained language models such as BERT and DialoGPT (Ghazarian et al., 2020; Mehri and Eskenazi, 2020a,b).

In this paper, we focus on predicting user sat-

isfaction ratings for the whole dialogue and compare various state-of-the-art regression methods. As mentioned earlier, one of our most important contributions is the introduction of Gaussian Process Regression (GPR) to the dialogue community as a means for dialogue system evaluation. GPR has been used before in the NLP community for machine translation quality estimation (Cohn and Specia, 2013) and emotion prediction (Beck et al., 2014). To our knowledge, in the dialogue community Gaussian Processes (i.e., the GP-SARSA algorithm) have only been used for dialogue policy learning via reinforcement learning (Gašić et al., 2010; Gašić and Young, 2014).

We compare GPR with Support Vector Regression (SVR), which is a strong baseline, and linear regression. Of course linear regression has been used before for dialogue evaluation (e.g., (Walker et al., 2000, 2001b; Georgila et al., 2019, 2020)). Classification based on Support Vector Machines has been used for interaction quality estimation (Ultes and Minker, 2014; Schmitt and Ultes, 2015).

3 Data and Features

The corpus that we use was derived from the original COMMUNICATOR corpus via automatic annotation (Georgila et al., 2005b, 2009). The original COMMUNICATOR corpus contained speech act and task annotations for the system's side of the conversation based on the DATE scheme (Walker and Passoneau, 2001). Georgila et al. (2005b, 2009) added speech act and task annotations for the user's side of the conversation, as well as information about the dialogue context, e.g., filled slots, filled slots values, grounded slots, etc. Georgila et al. (2005b, 2009) present in detail how these fully automatic annotations were generated. Figure 2 shows an example dialogue excerpt including speech act and task annotations, and Figure 3 depicts an example dialogue state corresponding to the dialogue status after user utterance 4 in Figure 2.

The automatic annotations were evaluated with respect to the task completion metrics of the original corpus and in comparison to hand-annotated data, which has verified their validity and reliability (Georgila et al., 2009). Over the years the utility of this annotated corpus has been demonstrated by its use by various researchers for different purposes, mainly, learning dialogue policies (Henderson et al., 2005; Frampton and Lemon, 2006; Henderson et al., 2008) and building simulated users (Schatzmann et al., 2005; Georgila et al., 2005a, 2006). More recently, it was used for system dialogue act selection for pre-training of goal-oriented dialogue policies (McLeod et al., 2019).

The dialogue context annotations are divided into 2 broad categories: logs of the current status of the slots (i.e., 'FilledSlotsStatus', 'FilledSlotsValuesStatus', 'GroundedSlotsStatus'), and logs containing information about how the status of the slots has changed over the course of the dialogue (i.e., 'FilledSlotsHist', 'FilledSlotsValuesHist', 'GroundedSlotsHist'). Because the former inform us about the current status of the slots they may only contain one instance per slot. The latter give us information about the order in which slots have been filled or confirmed and may contain several instances of the same slot, e.g., a slot could be confirmed twice. Thus, if a confirmed slot is refilled with a new value it will remain in the 'ConfirmedSlotsHist' field even though its new value has not been confirmed yet. The history of speech acts and tasks is also included in the annotations.

The annotated corpus (COMMUNICATOR 2001 part) consists of 1683 dialogues collected between human users and 8 dialogue systems but for our experiments we only used dialogues for which all user ratings were available: ATT (157 dialogues), BBN (137 dialogues), CMU (69 dialogues), COLORADO (157 dialogues), IBM (77 dialogues), LUCENT (140 dialogues), MIT (166 dialogues), and SRI (103 dialogues). The first half of the dialogues from each system are included in the training data set (500 dialogues) and the rest are included in the test data set (506 dialogues).

We extract 16 features from this corpus and perform regression experiments in order to predict the following user satisfaction ratings on a Likert scale (1-5, higher is better): ease of the tasks the user had to accomplish (henceforth referred to as 'Task-Ease'), whether it was easy or not to understand the system (henceforth referred to as 'System-Comprehend-Ease'), the user's expertise (henceforth referred to as 'User-Expertise'), whether the system behaved as expected (henceforth referred to as 'System-Behaved-As-Expected'), and if the user would use the system again in the future or not (henceforth referred to as 'System-Future-Use'). We use 16 features divided into 4 categories:

- **duration-related features (9):** overall duration, duration of the system talking part, duration of the user talking part, overall average

duration per utterance, average duration per system utterance, average duration per user utterance, number of overall speech acts, number of system speech acts, number of user speech acts;

- **slots-related features (3):** number of filled slots, number of filled slots without any 'null' values, number of grounded slots (all at the end of the dialogue);
- **slots-history-related features (3):** number of filled slots in the dialogue history, number of filled slots without any 'null' values in the dialogue history, number of grounded slots in the dialogue history (all at the end of the dialogue);
- **word error rate (WER) (1)**: calculated by comparing the speech recognition output to the transcription of the user utterance (this information was included in the original COMMUNICATOR corpus).

We remove all empty ('[]') values, and also distinguish between slots filled with normal versus 'null' values as an extra piece of information (see Figure 3 in the Appendix). Because we only consider numbers of slots, speech acts, and tasks, and not their specific types or values, our features are domain-independent and also automatically extracted from the data. We replace feature values with z-scores, i.e., from each feature value we subtract the mean for that feature and then divide by the standard deviation for that feature. For each feature, the mean and standard deviation are calculated on the training data set.

4 Regression Methods

For our experiments we use various regression methods, specifically, linear regression, linear regression with L2 regularization (also known as linear ridge regression), Support Vector Regression (SVR), and Gaussian Process Regression (GPR). As mentioned above, to our knowledge, GPR has not been used before for dialogue system evaluation, even though GPR is considered as the state-of-the-art for regression and is continually attracting more and more interest.

Gaussian Processes (GPs) are an elegant framework for probabilistic inference incorporating kernels and Bayesian inference (Rasmussen and Williams, 2006). A GP is a probability distribution over possible functions that fit a set of data points. GPs are similar to Support Vector Machines in the sense that they use kernels for non-linear modelling. The main difference is that GPs are probabilistic models and support exact Bayesian inference for regression; approximate inference is required for classification (Rasmussen and Williams, 2006). GPs are also more flexible in terms of fitting the kernel hyperparameters even for complex composite kernels. Because of their probabilistic formulation GPs can also be incorporated into larger graphical models and explicitly model uncertainty.

A kernel is a way of computing the dot product of two vectors in a high dimensional feature space. Thus the kernel function $k(x_i, x_j)$ essentially tells the model how similar two data points (x_i, x_j) are.

For SVR we use scikit-learn[1]. For GPR we use the GPy library[2]. For SVR we experimented with various kernels but using the RBF (radial basis function) kernel resulted in the best performance. For GPR we use the exponential kernel, the rational quadratic kernel, the RBF kernel, the sum of the exponential and the periodic kernel, the sum of the rational quadratic and the periodic kernel, and the sum of the RBF and the periodic kernel.

The RBF kernel is also called the exponentiated quadratic kernel, the squared exponential kernel, or the Gaussian kernel. The rational quadratic kernel is equivalent to adding together multiple RBF kernels with various length scales. For all GPR experiments we varied the length scale and we report results for length scale equal to 1 (the higher the value of the length scale the smoother the learned function). Varying the length scale did not result in significant differences. Note that adding two kernels can be thought of as an OR operation. Thus, the resulting kernel will have a high value if either of the two base kernels has a high value.

All of the above are frequently used kernels for GPR that seem to perform well for various types of data. Training custom kernels may lead to better results but this is a complex process and one of our future work directions. Note that we also experimented with other kernels such as the Matérn 3/2 and 5/2 kernels (Rasmussen and Williams, 2006) as well as the periodic kernel by itself but we do not report these results due to space restrictions. These kernels performed consistently worse.

[1] https://scikit-learn.org/stable/
[2] https://gpy.readthedocs.io/en/deploy/

	linear	linear ridge	SVR RBF	GPR exp	GPR ratq	GPR RBF	GPR exp+per	GPR ratq+per	GPR RBF+per
Task-Ease									
RMSE	1.428	1.376	1.303	1.279	1.281	1.434	1.278	1.281	**1.277**
r	0.349	0.373	0.477	0.493	0.491	0.298	0.494	0.491	**0.498**
ρ	0.425	0.435	0.48	0.506	0.501	0.322	**0.507**	0.501	0.502
System-Comprehend-Ease									
RMSE	1.302	1.242	1.203	**1.161**	1.165	1.246	1.168	1.165	1.178
r	0.161	0.2	0.354	**0.383**	0.378	0.197	0.374	0.378	0.356
ρ	0.242	0.257	0.366	**0.391**	0.383	0.194	0.378	0.383	0.366
User-Expertise									
RMSE	1.405	1.359	1.305	1.297	1.294	1.317	1.297	1.294	**1.283**
r	0.137	0.156	**0.272**	0.252	0.248	0.174	0.253	0.248	0.268
ρ	0.184	0.184	**0.281**	0.266	0.258	0.148	0.267	0.258	0.276
System-Behaved-As-Expected									
RMSE	1.397	1.38	1.295	1.282	1.288	1.419	**1.274**	1.288	1.288
r	0.321	0.333	0.44	0.453	0.447	0.343	**0.462**	0.447	0.445
ρ	0.377	0.382	0.443	0.454	0.451	0.395	**0.465**	0.451	0.451
System-Future-Use									
RMSE	1.492	1.455	**1.397**	1.398	1.398	1.48	1.41	1.398	1.41
r	0.251	0.269	**0.382**	0.376	0.375	0.256	0.342	0.375	0.343
ρ	0.281	0.285	**0.379**	0.362	0.364	0.254	0.333	0.364	0.339

Table 1: Results for RMSE, Pearson's r correlation, and Spearman's ρ correlation, for various regression methods using all the training data and all features; "exp" stands for exponential, "ratq" for rational quadratic, and "per" for periodic kernel. The best values are shown in bold.

5 Results

To measure the predictive power of our models we compare the predictions of each model for each of the 5 user ratings with the ground truth, i.e., the ratings in the test data. We calculate the Root Mean Square Error (RMSE), Pearson's r correlation, and Spearman's ρ correlation.

RMSE measures the average error between the model predictions and the ground truth and its value varies from 0 to 4, given that user ratings were on a scale from 1 to 5. Lower RMSE values are better.

Pearson's r measures the linear relationship between the model predictions and the ground truth and can range from -1 to 1 (the higher the better).

Spearman's ρ is based on the ranked values of the ratings rather than the raw data, which makes sense in our case given that the user ratings can be thought of as some kind of ranking between interactions even though users rated individual interactions. Spearman's ρ determines the degree to which the relationship between the compared variables is monotonic. Spearman's ρ ranges from -1 to 1 (the higher the better).

5.1 Which regression method works best?

Table 1 shows the RMSE, r, and ρ values for the regression methods and kernel types mentioned in Section 4. Here we use all the training data and all features. Clearly SVR and GPR outperform linear and linear ridge regression. For all rating types, GPR results in modest gains compared to SVR, except for 'System-Future-Use'. For 'User-Expertise' SVR results in higher correlation scores than GPR but also higher RMSE. As we will see later, the gains resulting from GPR (compared to SVR) are statistically significant mainly for 'Task-Ease' and 'System-Comprehend-Ease'. For GPR the exponential and rational quadratic kernels outperform the RBF kernel. Adding the periodic kernel to the exponential, rational quadratic, and RBF kernels respectively may lead to improved performance. Adding the exponential and the periodic kernel results in slight gains for 'Task-Ease', 'User-Expertise', and 'System-Behaved-As-Expected'. Adding the rational quadratic and the periodic kernel did not make any difference compared to just using the rational quadratic kernel. Adding the RBF and the periodic kernel led to improved values.

	linear	linear ridge	SVR RBF	GPR exp	GPR ratq	GPR RBF	GPR exp+per	GPR ratq+per	GPR RBF+per
Task-Ease									
20%	1.794	1.446	1.382	1.366	1.479	1.479	1.364	1.364	1.364
40%	1.56	1.415	1.382	1.359	1.349	1.385	1.393	1.39	1.417
60%	1.44	1.397	1.348	1.332	1.331	1.373	1.341	1.347	1.347
80%	1.414	1.369	1.296	1.278	**1.277**	1.31	1.283	1.281	1.305
100%	1.428	1.376	1.303	1.279	1.281	1.434	1.278	1.281	**1.277**
System-Comprehend-Ease									
20%	1.886	1.508	1.249	1.209	1.265	1.265	1.204	1.199	1.199
40%	1.65	1.449	1.222	1.219	1.212	1.312	1.228	1.222	1.231
60%	1.331	1.25	1.188	1.189	1.18	1.241	1.189	1.222	1.222
80%	1.262	1.228	1.172	1.159	**1.154**	1.199	1.161	1.161	1.185
100%	1.302	1.242	1.203	1.161	1.165	1.246	1.168	1.165	1.178
User-Expertise									
20%	1.535	1.499	1.312	1.315	1.329	1.329	1.315	1.309	1.309
40%	1.48	1.395	1.33	1.321	1.319	1.352	1.369	1.399	1.399
60%	1.461	1.418	1.353	1.346	1.34	1.384	1.342	1.366	1.366
80%	1.397	1.361	1.326	1.307	1.299	1.342	1.304	1.31	1.325
100%	1.405	1.359	1.305	1.297	1.294	1.317	1.297	1.294	**1.283**
System-Behaved-As-Expected									
20%	1.777	1.397	1.379	1.431	1.432	1.432	1.431	1.346	1.349
40%	1.506	1.338	1.385	1.333	1.334	1.33	1.422	1.414	1.403
60%	1.404	1.34	1.355	1.305	1.309	1.332	1.313	1.328	1.328
80%	1.383	1.337	1.316	1.287	1.288	1.314	1.29	1.279	1.281
100%	1.397	1.38	1.295	1.282	1.288	1.419	**1.274**	1.288	1.288
System-Future-Use									
20%	1.847	1.643	1.592	1.541	1.541	1.541	1.541	1.5	1.5
40%	1.742	1.524	1.558	1.506	1.47	1.486	1.468	1.494	1.5
60%	1.542	1.489	1.456	1.444	1.443	1.463	1.44	1.444	1.444
80%	1.498	1.461	1.438	1.407	1.41	1.424	1.404	1.411	1.411
100%	1.492	1.455	**1.397**	1.398	1.398	1.48	1.41	1.398	1.41

Table 2: Results for RMSE, for various regression methods using all features, and varying the percentage of training data (20%, 40%, 60%, 80%, 100%); "exp" stands for exponential, "ratq" for rational quadratic, and "per" for periodic kernel. The best values are shown in bold.

5.2 What is the impact of varying the training data size?

Table 2 shows the RMSE values for the regression methods and kernel types mentioned in Section 4. We use all features but vary the percentage of training data (20%, 40%, 60%, 80%, 100%, from each system respectively). Due to space constraints we do not report results on correlation. The values of Pearson's r and Spearman's ρ are consistent with the corresponding RMSE values (the lower the RMSE the higher the correlation).

As expected, for most rating types and methods the larger the size of the training data set the better the performance. However, there are some exceptions when we move from using 80% of the training data to 100% of the training data.

For 'Task-Ease' and for the GPR cases when we add the periodic kernel to the exponential, rational quadratic, and RBF kernels respectively, performance improves or remains stable when we use 100% of the training data but in all other cases it drops. For 'System-Comprehend-Ease', performance improves when we use 100% of the training data only for the GPR case with the sum of the RBF kernel and periodic kernel. For 'User-Expertise' using 100% of the training data outperforms using 80% of the data for all cases except for linear regression. For 'System-Behaved-As-Expected' some-

times adding the last 20% of the data helps but not always. It does not help for linear and linear ridge regression, GPR with the rational quadratic kernel, GPR with the RBF kernel, GPR with the sum of the rational quadratic and periodic kernel, and GPR with the sum of the RBF and periodic kernel. For 'System-Future-Use' using 80% of the training data is better than using 100% of the training data for GPR with the RBF kernel and GPR with the sum of the exponential and periodic kernel. Thus, we can see that in some cases some kind of over-fitting takes place as we add more data.

5.3 Which feature combinations work best?

Table 3 shows the RMSE, r, and ρ values for the regression methods SVR with the RBF kernel and GPR with the exponential kernel. Here we use all the training data but vary the features.

Tables 1 and 2 show that there is not much difference between using GPR with the exponential kernel and GPR with the rational quadratic kernel or their counterparts with the addition of the periodic kernel. For this reason and because of space limitations, for the third research question, we only consider GPR with the exponential kernel and SVR with the RBF kernel. So far we have seen that in many cases GPR outperforms SVR (a strong baseline) but here we also want to see if this is the case for different feature combinations and report on statistical significance.

In terms of feature combinations we get the best results when we use all features except for 'System-Comprehend-Ease' and 'User-Expertise'. As we can see from the first two rows for each rating type, sometimes the duration features are more predictive than the slot features, and vice versa. Combining these features leads to further improvements for all rating types and both SVR and GPR. Adding WER to duration features (dur+WER) always helps except for 'User-Expertise'. Adding slots features to duration features and WER (dur+WER+sl) also always helps. Adding slots history features to WER, slots, and duration features (which is equivalent to using all features) helps in most cases except for 'System-Comprehend-Ease' and 'User-Expertise'. When we remove WER from all features (all-WER) performance improves slightly for 'System-Comprehend-Ease' with SVR, and 'User-Expertise' with both SVR and GPR.

Regarding comparing SVR and GPR, for 'Task-Ease' and 'System-Comprehend-Ease', GPR is almost always significantly better than SVR. For all statistical significance calculations, for comparing SVR and GPR, we use the squared error values and the Wilcoxon signed-rank test with Holm-Bonferroni correction for repeated measures. For 'User-Expertise' and 'System-Behaved-As-Expected', GPR is significantly better than SVR when we use the slots features ($p < 0.01$ and $p < 0.001$ respectively). For 'System-Future-Use', differences between SVR and GPR performance are not significant.

Walker et al. (2001b) also showed the importance of duration and WER for user satisfaction prediction using the original COMMUNICATOR corpus. WER cannot be available unless the user speech is transcribed so an alternative approach would be to use speech recognition confidence scores as a proxy for WER. We also present results assuming that the user's perceived task completion is available (as a high bar for prediction), and as expected, this extra piece of information can significantly improve performance ($p < 0.001$).

We also implemented 5 simple baselines where the model always predicts the same score. Thus, Baseline 1 always predicts the score 1, Baseline 2 always predicts 2, etc. Table 4 shows results for RMSE for the baseline that always predicts the score 3 and the majority baseline for each type of rating, and the best performance of GPR with the exponential kernel (based on Table 3). Figure 1 shows the distributions of values (1 to 5) for each type of rating. The distributions in the training and test data differ, and each type of rating follows different patterns. Based on the distributions for the training data, Baseline 4 is equivalent to the majority baseline for 'Task-Ease', 'System-Comprehend-Ease', 'User-Expertise', and 'System-Behaved-As-Expected', and Baseline 1 is the majority baseline for 'Future-Use'. Baseline 3 generates RMSE values of approximately 1.5 and the only case where the majority baseline works well is for 'System-Comprehend-Ease'. Differences in performance between GPR and all baselines for all rating types are statistically significant ($p < 0.001$).

6 Conclusion

We used regression for predicting user ratings of their interaction with a dialogue system using the richly annotated version of the COMMUNICATOR corpus (Georgila et al., 2005b, 2009). We explored 3 research questions: (i) Which regression method

	SVR-RMSE	SVR-r	SVR-ρ	GPR-RMSE	GPR-r	GPR-ρ	Stat Sign
Task-Ease							
dur	1.37	0.408	0.417	1.319	0.442	0.443	$p < 0.01$
sl	1.377	0.395	0.374	1.357	0.385	0.385	$p < 0.01$
dur+sl	1.334	0.452	0.448	1.292	0.476	0.484	$p < 0.01$
dur+WER	1.327	0.452	0.459	1.292	0.483	0.495	$p < 0.05$
dur+WER+sl	1.316	0.466	0.469	1.281	0.491	0.503	$p < 0.01$
all-WER	1.311	0.473	0.472	1.287	0.484	0.495	$p < 0.05$
all	1.303	0.477	0.48	**1.279**	**0.493**	**0.506**	$p < 0.05$
all+PTC	1.166	0.61	0.605	1.145	0.627	0.636	$p < 0.05$
System-Comprehend-Ease							
dur	1.231	0.269	0.314	1.187	0.339	0.341	$p < 0.01$
sl	1.208	0.349	0.353	1.203	0.343	0.306	n.s.
dur+sl	1.191	0.377	0.382	1.16	0.387	0.387	$p < 0.01$
dur+WER	1.229	0.293	0.318	1.178	0.368	0.376	$p < 0.001$
dur+WER+sl	1.202	0.359	0.364	**1.157**	**0.393**	**0.396**	$p < 0.001$
all-WER	1.191	0.373	0.379	1.162	0.377	0.38	$p < 0.05$
all	1.203	0.354	0.366	1.161	0.383	0.391	$p < 0.001$
all+PTC	1.192	0.386	0.397	1.137	0.434	0.439	$p < 0.001$
User-Expertise							
dur	1.312	0.26	0.29	1.287	0.25	0.262	n.s.
sl	1.317	0.223	0.187	1.305	0.191	0.164	$p < 0.01$
dur+sl	**1.28**	**0.306**	**0.314**	**1.28**	0.275	0.288	n.s.
dur+WER	1.313	0.25	0.27	1.288	0.248	0.263	n.s.
dur+WER+sl	1.287	0.295	0.295	1.283	0.27	0.278	n.s.
all-WER	1.3	0.28	0.3	1.296	0.26	0.274	n.s.
all	1.305	0.272	0.281	1.297	0.252	0.266	n.s.
all+PTC	1.289	0.293	0.315	1.276	0.297	0.325	n.s.
System-Behaved-As-Expected							
dur	1.341	0.398	0.401	1.333	0.392	0.385	n.s.
sl	1.417	0.328	0.307	1.363	0.331	0.322	$p < 0.001$
dur+sl	1.301	0.442	0.436	1.294	0.439	0.432	n.s.
dur+WER	1.309	0.424	0.426	1.301	0.429	0.435	n.s.
dur+WER+sl	1.298	0.439	0.441	1.283	**0.453**	0.453	n.s.
all-WER	1.295	0.447	0.442	1.288	0.446	0.443	n.s.
all	1.295	0.44	0.443	**1.282**	**0.453**	**0.454**	n.s.
all+PTC	1.191	0.568	0.573	1.185	0.572	0.577	n.s.
System-Future-Use							
dur	1.445	0.307	0.298	1.416	0.338	0.323	n.s.
sl	1.446	0.266	0.265	1.446	0.315	0.322	n.s.
dur+sl	1.415	0.357	0.356	1.4	0.364	0.35	n.s.
dur+WER	1.422	0.341	0.333	1.403	0.372	0.364	n.s.
dur+WER+sl	1.41	0.364	0.364	**1.397**	0.374	0.363	n.s.
all-WER	1.405	0.367	0.365	**1.397**	0.37	0.355	n.s.
all	**1.397**	**0.382**	**0.379**	1.398	0.376	0.362	n.s.
all+PTC	1.31	0.489	0.485	1.321	0.49	0.481	n.s.

Table 3: Results for RMSE, Pearson's r correlation, and Spearman's ρ correlation, for SVR with the RBF kernel and GPR with the exponential kernel using all the training data and varying feature combinations; "dur" stands for duration, "sl" for slots, and "PTC" for perceived task completion. The best values are shown in bold. The last column shows statistical significance ("n.s." stands for non-significant).

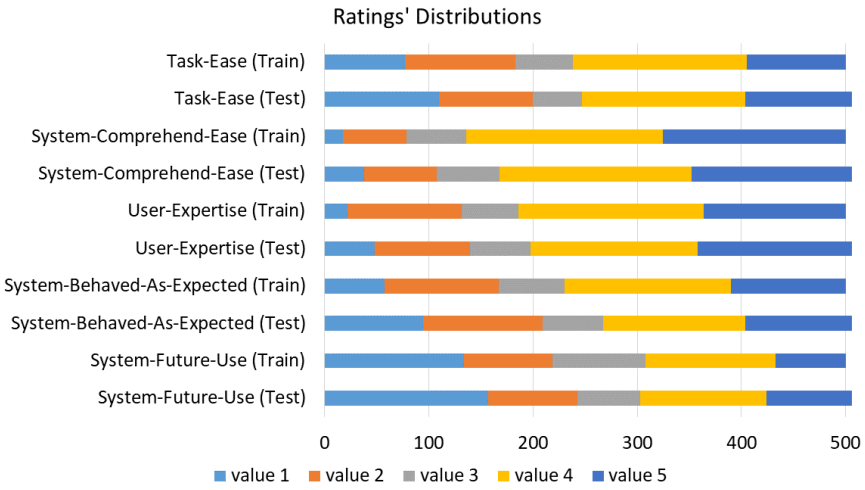

Figure 1: Ratings' distributions for the training and test data.

	Bas 3	Bas maj	GPR
Task-Ease	1.471	1.721	**1.279**
System-Comprehend-Ease	1.421	1.285	**1.157**
User-Expertise	1.431	1.41	**1.28**
System-Behaved-As-Expected	1.433	1.705	**1.282**
System-Future-Use	1.516	2.321	**1.397**

Table 4: Results for RMSE for the baselines and the best performance of GPR with the exponential kernel based on Table 3. The best values are shown in bold.

works best and does the choice of kernel matter for kernel-based regression? (i) What is the impact of varying the training data size? (iii) Which feature combinations work best?

To answer the first question we compared various state-of-the-art regression methods: linear regression, linear ridge regression, SVR, and GPR. We also varied the kernel type for GPR. To our knowledge, GPR has never been used before for dialogue system evaluation (or generally by the dialogue community) despite the fact that it is considered as the state-of-the-art for regression in other research areas. In many cases (mainly for 'Task-Ease' and 'System-Comprehend-Ease'), GPR led to modest but statistically significant gains compared to SVR (a strong baseline), and the type of kernel used mattered. The gains were even larger when compared to linear regression.

To answer the second question we varied the training data size and reported on its impact on performance for all regression methods. The larger the training set the higher the gains but for some methods more data may result in over-fitting.

To answer the third question we varied the feature combinations used for regression and showed how the choice of features affects the prediction quality of our models. Even though the features we used are domain-independent, our experiments provided valuable insights about the benefits of different feature combinations, including features taking into account dialogue context and dialogue history, as well as feature combinations that do not rely on complex annotations. Some feature combinations worked better than others but in most cases the best results were obtained with all features.

Overall the RMSE ranged roughly from 1 to 1.5 depending on the regression method and kernel type, training data size, and feature combination. Predicting individual user judgements is a hard task (Möller and Ward, 2008), and given that we did not use any domain-dependent features our results are promising. For future work we will train custom kernels and measure if performance improves. We also expect performance gains from using domain-dependent features.

Acknowledgements

This work was partly supported by the U.S. Army. Statements and opinions expressed and content included do not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

Daniel Beck, Trevor Cohn, and Lucia Specia. 2014. Joint emotion analysis via multi-task Gaussian Processes. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1798–1803, Doha, Qatar.

Trevor Cohn and Lucia Specia. 2013. Modelling annotator bias with multi-task Gaussian Processes: An application to machine translation quality estimation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 32–42, Sofia, Bulgaria.

Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2021. Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, 54:755–810.

Klaus-Peter Engelbrecht, Florian Gödde, Felix Hartard, Hamed Ketabdar, and Sebastian Möller. 2009. Modeling user satisfaction with hidden Markov models. In *Proc. of the Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 170–177, London, UK.

Mary Ellen Foster, Manuel Giuliani, and Alois Knoll. 2009. Comparing objective and subjective measures of usability in a human-robot dialogue system. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 879–887, Suntec, Singapore.

Matthew Frampton and Oliver Lemon. 2006. Learning more effective dialogue strategies using limited dialogue move features. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 185–192, Sydney, Australia.

Michel Galley, Chris Brockett, Alessandro Sordoni, Yangfeng Ji, Michael Auli, Chris Quirk, Margaret Mitchell, Jianfeng Gao, and Bill Dolan. 2015. DeltaBLEU: A discriminative metric for generation tasks with intrinsically diverse targets. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)– Short Papers*, pages 445–450, Beijing, China.

Milica Gašić, Filip Jurčíček, Simon Keizer, Francois Mairesse, Blaise Thomson, Kai Yu, and Steve Young. 2010. Gaussian Processes for fast policy optimisation of POMDP-based dialogue managers. In *Proceedings of the Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 201–204, Tokyo, Japan.

Milica Gašić and Steve Young. 2014. Gaussian Processes for POMDP-based dialogue manager optimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1):28–40.

Kallirroi Georgila, Carla Gordon, Hyungtak Choi, Jill Boberg, Heesik Jeon, and David Traum. 2019. Toward low-cost automated evaluation metrics for Internet of Things dialogues. In *Proc. of the International Workshop on Spoken Dialogue Systems Technology (IWSDS), Lecture Notes in Electrical Engineering 579*, pages 161–175, Singapore.

Kallirroi Georgila, Carla Gordon, Volodymyr Yanov, and David Traum. 2020. Predicting ratings of real dialogue participants from artificial data and ratings of human dialogue observers. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, pages 726–734, Marseille, France (Online).

Kallirroi Georgila, James Henderson, and Oliver Lemon. 2005a. Learning user simulations for Information State Update dialogue systems. In *Proc. of Interspeech*, pages 893–896, Lisbon, Portugal.

Kallirroi Georgila, James Henderson, and Oliver Lemon. 2006. User simulation for spoken dialogue systems: Learning and evaluation. In *Proc. of Interspeech*, pages 1065–1068, Pittsburgh, Pennsylvania, USA.

Kallirroi Georgila, Oliver Lemon, and James Henderson. 2005b. Automatic annotation of COMMUNICATOR dialogue data for learning dialogue strategies and user simulations. In *Proc. of the Workshop on the Semantics and Pragmatics of Dialogue (SemDial:DIALOR)*, pages 61–68, Nancy, France.

Kallirroi Georgila, Oliver Lemon, James Henderson, and Johanna D. Moore. 2009. Automatic annotation of context and speech acts for dialogue corpora. *Journal of Natural Language Engineering*, 15(3):315–353.

Sarik Ghazarian, Ralph Weischedel, Aram Galstyan, and Nanyun Peng. 2020. Predictive engagement: An efficient metric for automatic evaluation of open-domain dialogue systems. In *Proc. of the AAAI Conference on Artificial Intelligence*, pages 7789–7796, New York, New York, USA.

Fenfei Guo, Angeliki Metallinou, Chandra Khatri, Anirudh Raju, Anu Venkatesh, and Ashwin Ram. 2017. Topic-based evaluation for conversational bots. In *Proc. of NIPS Workshop on Conversational AI: Today's Practice and Tomorrow's Potential*, Long Beach, California, USA.

Helen Hastie. 2012. Metrics and evaluation of spoken dialogue systems. In Oliver Lemon and Olivier Pietquin, editors, *Data-Driven Methods for Adaptive Spoken Dialogue Systems*, pages 131–150. Springer.

James Henderson, Oliver Lemon, and Kallirroi Georgila. 2005. Hybrid reinforcement/supervised learning for dialogue policies from COMMUNICATOR data. In *Proc. of the IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 68–75, Edinburgh, UK.

James Henderson, Oliver Lemon, and Kallirroi Georgila. 2008. Hybrid reinforcement/supervised learning of dialogue policies from fixed datasets. *Computational Linguistics*, 34(4):487–511.

Ryuichiro Higashinaka, Yasuhiro Minami, Kohji Dohsaka, and Toyomi Meguro. 2010. Issues in predicting user satisfaction transitions in dialogues: Individual differences, evaluation criteria, and prediction models. In *Proc. of the International Workshop on Dialogue Systems Technology (IWSDS), Lecture Notes in Computer Science 6392*, pages 48–60, Gotemba, Shizuoka, Japan.

Kate S. Hone and Robert Graham. 2000. Towards a tool for the Subjective Assessment of Speech System Interfaces (SASSI). *Journal of Natural Language Engineering*, 6(3-4):287–303.

Chia-Wei Liu, Ryan Lowe, Iulian V. Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2122–2132, Austin, Texas, USA.

Ryan Lowe, Iulian V. Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. On the evaluation of dialogue systems with next utterance classification. In *Proc. of the Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 264–269, Los Angeles, California, USA.

Sarah McLeod, Ivana Kruijff-Korbayová, and Bernd Kiefer. 2019. Multi-task learning of system dialogue act selection for supervised pretraining of goal-oriented dialogue policies. In *Proc. of the Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 411–417, Stockholm, Sweden.

Shikib Mehri, Jinho Choi, Luis Fernando D'Haro, Jan Deriu, Maxine Eskenazi, Milica Gasic, Kallirroi Georgila, Dilek Hakkani-Tur, Zekang Li, Verena Rieser, Samira Shaikh, David Traum, Yi-Ting Yeh, Zhou Yu, Yizhe Zhang, and Chen Zhang. 2022. Report from the NSF Future Directions Workshop on Automatic Evaluation of Dialog: Research Directions and Challenges. In *arXiv 2203.10012*.

Shikib Mehri and Maxine Eskenazi. 2020a. Unsupervised evaluation of interactive dialog with DialoGPT. In *Proc. of the Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 225–235, Online.

Shikib Mehri and Maxine Eskenazi. 2020b. USR: An unsupervised and reference free evaluation metric for dialog generation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 681–707, Online.

Sebastian Möller and Nigel Ward. 2008. A framework for model-based evaluation of spoken dialog systems. In *Proc. of the Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 182–189, Columbus, Ohio, USA.

Carl Edward Rasmussen and Christopher K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. MIT Press.

Susan Robinson, Antonio Roque, and David Traum. 2010. Dialogues in context: An objective user-oriented evaluation approach for virtual human dialogue. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, pages 64–71, Valletta, Malta.

Jost Schatzmann, Kallirroi Georgila, and Steve Young. 2005. Quantitative evaluation of user simulation techniques for spoken dialogue systems. In *Proc. of the Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 45–54, Lisbon, Portugal.

Alexander Schmitt and Stefan Ultes. 2015. Interaction Quality: Assessing the quality of ongoing spoken dialog interaction by experts—And how it relates to user satisfaction. *Speech Communication*, 74:12–36.

David R. Traum, Susan Robinson, and Jens Stephan. 2004. Evaluation of multi-party virtual reality dialogue interaction. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, pages 1699–1702, Lisbon, Portugal.

Stefan Ultes and Wolfgang Minker. 2014. Interaction quality estimation in spoken dialogue systems using hybrid-HMMs. In *Proc. of the Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 208–217, Philadelphia, Pennsylvania, USA.

Marilyn Walker, J. Aberdeen, J. Boland, E. Bratt, J. Garofolo, Lynette Hirschman, A. Le, S. Lee, Shrikanth Narayanan, K. Papineni, Bryan Pellom, Joseph Polifroni, Alexandros Potamianos, P. Prabhu, Alexander I. Rudnicky, G. Sanders, Stephanie Seneff, D. Stallard, and Steve Whittaker. 2001a. DARPA Communicator dialog travel planning systems: the June 2000 data collection. In *Proc. of Interspeech/Eurospeech*, pages 1371–1374, Aalborg, Denmark.

Marilyn Walker, Candace Kamm, and Diane Litman. 2000. Towards developing general models of usability with PARADISE. *Journal of Natural Language Engineering*, 6(3-4):363–377.

Marilyn Walker and Rebecca Passoneau. 2001. DATE: A dialogue act tagging scheme for evaluation of spoken dialogue systems. In *Proc. of the Human Language Technologies Conference*, pages 1–8, San Diego, California, USA.

Marilyn A. Walker, Rebecca Passoneau, and Julie E. Boland. 2001b. Quantitative and qualitative evaluation of Darpa Communicator spoken dialogue systems. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 515–522, Toulouse, France.

Helen Wright-Hastie, Rashmi Prasad, and Marilyn Walker. 2002. What's the trouble: Automatically identifying problematic dialogues in DARPA Communicator dialogue systems. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 384–391, Philadelphia, Pennsylvania, USA.

Appendix

System utterance 1	what trip would you like to take?
System speech act	request_info
System task	top_level_trip
User utterance 1 *ASR output*	i'd like to fly from boston to detroit on november second
User transcription	i'd like to fly from boston to detroit on november second
User speech act	[provide_info,provide_info,provide_info]
User task	[orig_city,dest_city,depart_date]
System utterance 2	on friday, november second
System speech act	implicit_confirm
System task	depart_arrive_date
System utterance 3	flying to detroit
System speech act	implicit_confirm
System task	dest_city
System utterance 4	pardon me
System speech act	apology
System task	meta_slu_reject
System utterance 5	did you say flying from boston instead?
System speech act	explicit_confirm
System task	orig_city
User utterance 2 *ASR output*	yes boston to detroit on northwest airline
User transcription	yes from boston to detroit on northwest airlines
User speech act	[yes_answer,reprovide_info,reprovide_info,provide_info]
User task	[null,orig_city,dest_city,airline]
System utterance 6	flying from boston to detroit
System speech act	implicit_confirm
System task	orig_dest_city
System utterance 7	on northwest
System speech act	implicit_confirm
System task	airline
System utterance 8	instead
System speech act	tbc
System task	tbc
System utterance 9	about what time do you want to leave?
System speech act	request_info
System task	depart_arrive_time
User utterance 3 *ASR output*	early morning july twenty p. m.
User transcription	early morning to arrive around two p. m.
User speech act	[provide_info,provide_info]
User task	[depart_date,depart_time]
System utterance 10	leaving in the early morning
System speech act	implicit_confirm
System task	depart_arrive_time
System utterance 11	pardon me
System speech act	apology
System task	meta_slu_reject
System utterance 12	did you say on friday, july twentieth
System speech act	explicit_confirm
System task	depart_arrive_date
System utterance 13	instead?
System speech act	tbc
System task	tbc
User utterance 4 *ASR output*	no november second
User transcription	no november second
User speech act	[no_answer,provide_info]
User task	[null,depart_date]

Figure 2: An example dialogue excerpt between the system and the user. Both the automatic speech recognition (ASR) output and the transcription of the user's speech are shown. Speech act and task annotations are depicted for both the system and the user. Note that user speech act and task annotations, and dialogue context information are derived from the ASR output.

DIALOGUE LEVEL
Turn: user
TurnStartTime: 991948554.109
TurnEndTime: 991948559.296
TurnNumber: 4
Speaker: user
UtteranceStartTime: 991948554.109
UtteranceEndTime: 991948559.296
UtteranceNumber: 4
DialogueActType: user
ConvDomain: about_task
SpeechAct: [no_answer,provide_info]
AsrInput: no <date_time>november second</date_time>
TransInput: no <date_time>november second</date_time>
Output:
TASK LEVEL
Task: [null,depart_date]
FilledSlot: [null,depart_date]
FilledSlotValue: [no,november second]
GroundedSlot: []
LOW LEVEL
WordErrorRatenoins: 0.00
WordErrorRate: 0.00
SentenceErrorRate: 0.00
KeyWordErrorRate: 0.0
ComputeErrorRatesReturnValue: 0
HISTORY LEVEL
FilledSlotsStatus: [orig_city],[dest_city],[airline],[null],[null],[null],[depart_time],[null],[depart_date]
FilledSlotsValuesStatus: [boston],[detroit],[northwest],[boston],[detroit],[yes],[p m],[no],[november second]
GroundedSlotsStatus: [],[orig_city],[dest_city],[airline],[]
SpeechActsHist: request_info,[provide_info,provide_info,provide_info],implicit_confirm,implicit_confirm,apology,
explicit_confirm,[yes_answer,reprovide_info,reprovide_info,provide_info],implicit_confirm,implicit_confirm,tbc,
request_info,[provide_info,provide_info],implicit_confirm,apology,explicit_confirm,tbc,[no_answer,provide_info]
TasksHist: top_level_trip,[orig_city,dest_city,depart_date],depart_arrive_date,dest_city,meta_slu_reject,
orig_city,[null,orig_city,dest_city,airline],orig_dest_city,airline,tbc,
depart_arrive_time,[depart_date,depart_time],depart_arrive_time,meta_slu_reject,depart_arrive_date,tbc,[null,depart_date]
FilledSlotsHist: [orig_city,dest_city,depart_date],[null,null,null,airline],[depart_date,depart_time],
[null,depart_date]
FilledSlotsValuesHist: [boston,detroit,november second],[yes,boston,detroit,northwest],[july twenty,p m],
[no,november second]
GroundedSlotsHist: [],[orig_city,dest_city,depart_date],[orig_city,dest_city,airline],[]

Figure 3: An example dialogue state generated after user utterance 4 in Figure 2. Note that sometimes empty ('[]') and 'null' values are generated but they do not affect the accuracy of the slot values.