ORDINAL REGRESSION FOR INTERACTION QUALITY PREDICTION

Layla El Asri[1,2], *Hatim Khouzaimi*[1,3], *Romain Laroche*[1] *and Olivier Pietquin*[4*]

[1]Orange Labs Issy-les-Moulineaux, France
[2]UMI 2958 GeorgiaTech-CNRS, Francee
[3]Laboratoire d'Informatique d'Avignon-CERI, France
[4]University Lille 1, LIFL (UMR 8022 CNRS/Lille 1) - SequeL team, France

ABSTRACT

The automatic prediction of the quality of a dialogue is useful to keep track of a spoken dialogue system's performance and, if necessary, adapt its behaviour. Classifiers and regression models have been suggested to make this prediction. The parameters of these models are learnt from a corpus of dialogues evaluated by users or experts. In this paper, we propose to model this task as an ordinal regression problem. We apply support vector machines for ordinal regression on a corpus of dialogues where each system-user exchange was given a rate on a scale of 1 to 5 by experts. Compared to previous models proposed in the literature, the ordinal regression predictor has significantly better results according to the following evaluation metrics: Cohen's agreement rate with experts ratings, Spearman's rank correlation coefficient, and Euclidean and Manhattan errors.

Index Terms— Interactive Systems, Statistical Learning, Performance Evaluation

1. INTRODUCTION

Spoken Dialogue Systems (SDS) are a compound of several modules: Automatic Speech Recognition (ASR), natural language understanding, dialogue management, natural language generation and speech synthesis. An individual technical evaluation of its components is not sufficient to evaluate a system. Indeed, the behaviour emerging from the interaction between the modules must be evaluated to ensure system appropriateness, correctness and adequacy [1]. Moreover, a technical evaluation should be carried along with a usability evaluation to measure user acceptance. In this context, two kinds of evaluation campaigns are carried on SDS. The first kind asks the user to give a numerical rating and/or fill in a questionnaire about the dialogue after its end [2, 3, 4]. The second one involves experts instead of users [5, 6]. In both cases, ratings are meant for system performance tracking and if necessary, system behaviour adaptation. These evaluation campaigns are costly so it is desirable to build an automatic predictor of ratings to pursue online system performance tracking. The predicted ratings might also serve to learn an optimal behaviour for the system *via* Reinforcement Learning (RL) [7, 8, 9].

Several models have been proposed to predict users or experts ratings in function of dialogue features [2, 10, 6]. This problem has been considered both as a regression [11] and a classification [12] issue. This paper proposes ordinal regression as a bridge between classification and regression. Ordinal regression returns discrete labels but takes into account the ordering of the labels.

Ordinal Regression with Support Vector machines (SVOR) is compared to several regression and classification models on the LEGO corpus [13] which contains 200 annotated dialogues with CMU's Let's Go system [14]. Each system-user exchange was given an Interaction Quality (IQ) rating on a scale of 1 to 5 by three experts. It is shown on this corpus that SVOR significantly performs best on four metrics, both distance and correlation-based.

Section 2 reviews models previously proposed for users or experts ratings prediction. Section 3 explains the metrics used to compare the models. Section 4 then gives an overview of the LEGO corpus. Section 5 describes the models compared on LEGO and Section 6 presents the results of the test of the models.

2. RELATION TO PRIOR WORK

User satisfaction is one of the core components of SDS usability, as well as efficiency and effectiveness. Möller [15] defines usability as the "suitability of a system or service to fulfil the user's requirements. Includes effectiveness and efficiency of the system and results in user satisfaction". User satisfaction reflects user perception of dialogue features [16]. Measuring user satisfaction is made by asking the user to give a numerical rating and/or fill in a questionnaire after each dialogue [2, 3, 4, 17]. Another way to measure the performance of a system is to appeal to experts and ask them to listen to dialogues and give a rating to the system according to its management of the interaction [5, 6]. This technique enables to eliminate subjectivity biases related to the user's environment or the perception of aspects unrelated to dialogue management such as text-to-speech voice [15, 6]. This requires nevertheless that the experts are trained to avoid a great variability in their rating styles. In all that follows, users ratings will be designated as User Satisfaction (US) and experts ratings as Interaction Quality (IQ) [6].

Automatically predicting US or IQ is useful to keep track of a spoken dialogue system's performance and, if necessary, adapt its behaviour. This adaptation can be made according to rules inferred from the ratings. Witt [18] proposed to compute online the Caller Experience Metric (CEM), a score on a scale of 1 to 5 given by an expert. To do so, he defined the rule in Equation 1 where the events are ASR rejection, user time out, user intention misunderstood, user contradicting the system and user asking to be transferred. d is a discounting factor to reduce the importance of the furthest events in time. Witt learnt the weight of each event on a set of annotated dialogues with three different systems.

$$\text{CEM}_t = d \times \text{ CEM}_{t-1} + \text{ weight of the latest event} \tag{1}$$

System behaviour adaptation can also be done by applying a data-driven statistical method such as RL [19, 20, 21]. An RL agent com-

[*] The authors would like to thank the the Dialogue Systems Group at Ulm University, Germany for providing the LEGO corpus.

pares different strategies on the basis of a numerical reward function. This function should evaluate the quality of a dialogue strategy. Inferring this function from US or IQ [22, 23] is therefore an efficient way to define it. Finally, another possibility is to ask experts to give an IQ rating for each system-user exchange and then train a model to predict this rating at a dialogue turn level. The system can then adapt its behaviour during a dialogue if its estimation of IQ degrades. For instance, in such a case, the SDS might prompt contextual help messages [24], change its current dialogue strategy [25] or even transfer the call to a human operator [6].

To predict US or IQ on a dialogue or dialogue turn level, regression and classification models were proposed. Walker *et. al* [2] applied Multiple Linear Regression (MLR), modelling US as the optimisation of task success and minimisation of dialogue costs such as dialogue duration and ASR rejections. Evanini *et. al* [5] built a decision tree to predict IQ. Engelbrecht *et. al* [10] used Hidden Markov Models (HMM) to represent the evolution of US throughout a dialogue. Hara *et. al* [26] also tackled US prediction. They built an N-gram model from dialogue acts. Higashinaka *et. al* compared HMM and CRF to estimate turn-level IQ. For the same problem, Schmitt *et. al* [6] proposed Support Vector Machines (SVM).

Modelling IQ prediction as a pure classification issue does not enable to take into account the natural ordering of the scores. Indeed, it cannot take into consideration the fact that, for an actual rating of 1, predicting 2 is better than predicting 5. Regression techniques do not either explicitly account for the ordering as they only try to minimise some distance metric between the ratings and the predictions. In this paper, we propose SVOR for turn-level IQ prediction [27, 28]. Ordinal regression bridges the gap between classification and regression. It takes as input a set of labelled examples with naturally ordered labels and then builds a predictor in order to minimise the absolute deviation from the true labels while taking into account the ordering of the labels.

3. METRICS

SVOR is compared to other regression and classification models on the basis of several metrics. For RL-based systems, correctly ordering the system's dialogue strategies is important. Indeed, if turn-level IQ predictions are used as rewards, they must maintain the ordering of system actions that was induced by the actual experts ratings. RL frameworks such as the Ordinal Markov Decision Process introduced by Weng [29] can then be applied to learn an optimal behaviour. To measure models performance on this aspect, we use Spearman's rank correlation coefficient [30]. This coefficient measures the correlation between two rankings. Let $y = \{y_1, ..., y_n\}$ and $\hat{y} = \{\hat{y}_1, ..., \hat{y}_n\}$ be respectively the IQ ratings and estimations on a set of n system-user exchanges. Let $r(y) = \{r(y_1), ..., r(y_n)\}$ and $r(\hat{y}) = \{r(\hat{y}_1), ..., r(\hat{y}_n)\}$ be their corresponding rankings. For instance, if $y = \{1, 15, 3, 12, 27\}$ than $r(y) = \{1, 4, 2, 3, 5\}$. As recalled in Equation 2 Spearman's rank correlation coefficient ρ is equal to the correlation between the two rankings $r(y)$ and $r(\hat{y})$.

$$\rho(y, \hat{y}) = \frac{\sum_i (r(y_i) - \text{mean}(r(y))) \sum_i (r(\hat{y}_i) - \text{mean}(r(\hat{y})))}{\sqrt{\sum_i (r(y_i) - \text{mean}(r(y)))^2} \sqrt{\sum_i (r(\hat{y}_i) - \text{mean}(r(\hat{y})))^2}} \tag{2}$$

We also compare models on the distances they try to minimise, namely the Euclidean and Manhattan errors, given in Equations 3 and 4.

$$\text{Euclidean\_error}(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{3}$$

$$\text{Manhattan\_error}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{4}$$

The agreement with the experts is measured with Cohen's κ coefficient [31]. This coefficient computes the probability of agreement between y and $\hat{y}$ taking off the agreement that might occur by chance. In Equation 5, $P(a)$ is the probability of agreement and $P(ca)$ the probability of agreement by chance.

$$\kappa(y, \hat{y}) = \frac{P(a) - P(ca)}{1 - P(ca)} \tag{5}$$

4. THE LEGO CORPUS

CMU's Let's Go system provides local information on bus schedules [14]. Schmitt *et. al* [13] had 200 dialogues with this system evaluated by three experts. These experts were asked to give an IQ rating on a scale of 1 to 5 after each system-user exchange. In total, 5282 system-user exchanges were rated. Following Schmitt *et. al* [6], in our experiments, we used the median value of the three ratings provided by the experts.

In LEGO, each system-user exchange is described as a set of features that were automatically computed or manually annotated. The feature set contains parameters related to automatic speech recognition (confidence score, rejection,...), natural language understanding (user dialogue act, semantic parse,...) and dialogue management (system dialogue act). Following Schmitt *et. al* [6], we used quantifiable features such as number of ASR rejections on three different levels: the value for the current dialogue turn, the mean value up to the current dialogue turn and the mean over the last three exchanges. A complete list of the features can be found in [13].

For our models comparison, we only kept automatically computable features. This choice was motivated by the fact that if IQ prediction should be used for system adaptation, it should be computable online. Two features were not numerical: User Dialogue Act (UDA) and System Dialogue Act (SDA). UDA could take values among which UDA_CONFIRM_DEPARTURE, UDA_LINE_INFORMATION, and so on. In these cases, noting n_a the number of possible labels, we split the feature into $n_a - 1$ variables, each of which being a boolean. SDA was split into 27 variables and UDA, 21. In total, we kept 67 features. Feature values were centered and normalised.

The generalisability of the methods described in the following section was assessed with 10-fold cross validation.

5. MODELS

5.1. Multiple Linear Regression

To do MLR, features were selected one by one. We computed the single regression coefficient for each feature and then we kept the feature with the highest coefficient in absolute value[1]. Let us denote ϕ_i this most explanatory feature and β_i its corresponding coefficient. We then used the error $IQ - \beta_i \phi_i$ as objective value and we computed single regression coefficients for the remaining features. We repeated this process until the squared error was minimised.

[1] we recall that features were centered and normalised

Covariance type	$k(x, x')$
Linear	$x^T \Lambda^{-2} x'$
Rational quadratic	$\sigma^2(1 + \frac{1}{2\alpha}(x - x')^T \Lambda^{-2}(x - x'))^{-\alpha}$
Squared Exponential	$\exp(-\frac{1}{2}(x - x')^T \Lambda^{-2}(x - x'))$

Table 1. Covariance matrices for Gaussian processes regression.

Kernel function type	$k(x, x')$
Linear	$x^T x'$
Radial Basis Function	$\exp(-\gamma(x - x')^T(x - x'))$

Table 2. Kernel functions for SVM. In our experiment, γ was set to $\frac{1}{67}$, 67 being the number of features per sample point.

5.2. Multivariate Adaptive Regression Splines

MARS was introduced by Friedman [32]. It models IQ as a linear function of hinge functions of the form: $B_i(x) = max(0, x - s)$ or $B_i(x) = max(0, s - x)$ where s is a training sample. MARS was run with the ARES toolbox for Matlab[2]. The algorithm requires two parameters, the maximum number of hinge functions *maxFuncs* and *cubic*, which indicates whether cubic splines should be used to smooth the edges. Different combinations of these parameters were tried but no significant impact on the evaluation metrics was noticed. Hence, the default parameters were kept (recommended in [32]), maxFuncs was set to 21 and cubic splines were not used.

5.3. Gaussian Processes

A Gaussian Process (GP) is a set of random variables which joint distribution is a Gaussian [33]. To perform GP regression, y is modelled as: $y = \hat{y} + \Delta\hat{y}$ with $\Delta\hat{y}$ a centered Gaussian noise vector and $\hat{y}$ is a GP which mean and variance are to be determined. One advantage in using GP over other regression methods is that it returns an entire distribution over the possible values instead of a point-based decision rule. Another advantage is that it enables a Bayesian treatment of the data, inferring a posterior distribution from a prior belief and observations. Following Rasmussen and Williams [33], we defined a centered prior $GP(0, k(x, x))$ where k is the covariance function. Exact inference could then be performed from the training samples to compute the posterior distribution. We used the GPML Matlab library by Rasmussen and Nickisch[3]. Three covariance functions were tested: linear, rational quadratic and squared exponential. Their expressions are recalled in Table 1. The parameters Λ, σ and α were all learnt by automatic relevance determination [34]. Since this computation is very costly ($O(n^3)$ with n the number of training samples), we only kept in memory a dictionary of points as proposed by Engel [35]. This reduced computational load to $O(m^2 n)$ with m the dictionary size. Section 6 only presents the results with the squared exponential kernel as it was the most efficient.

5.4. Support Vector Machines

Originally, SVM [36] were meant for 2-class classification. They take a set of labelled examples $\{(x_i, y_i)\}$ where $y_i \in \{-1, 1\}$ and then they look for a hyperplane in a feature space that separates the two classes so that the minimal margin between the samples is maximised. The decision function returned is given in Equation 6 where $\phi(x)$ is the projection of x into the high dimensional feature space.

$$f(x) = \text{sign}(\omega^T \phi(x) + b) \tag{6}$$

SVM for Classification (SVMClass), Support Vector Machines for Regression (SVR) and SVOR solve a quadratic programming problem to find the weight vector ω and the constant b. The difference between the approaches lies in the constraints of the problem and the form of the decision function. These are detailed in the following sections.

5.4.1. *Support Vector Machines for Classification*

To perform 5-class classification with SVM, a one-versus-one approach was applied. 10 classifiers were learnt, each deciding between two classes. A new example was assigned the class that was the most voted for. Ties always favoured the lowest label as predicting a low IQ instead of a higher one is more critical than the other way round. The problem solved by each classifier is recalled in Equation 7. The decision function is the one of Equation 6.

$$\begin{aligned} &\min_{\omega,b,\xi} \frac{1}{2}\omega^T\omega + \sum_i \xi_i \\ &\text{subject to } y_i(\omega^T\phi(x_i) + b) > 1 - \xi_i \\ &\qquad \xi_i > 0 \;\forall\, i \end{aligned} \tag{7}$$

We used the LIBSVM software [37]. As showed in Table 2, two kernel functions were tested, a linear and a radial basis function. We only present the results with the radial basis kernel as it performed better.

5.4.2. *Support Vector Machines for Regression*

Drucker *et. al* [38] introduced SVR. The weights of the SVM are learnt to assure that the absolute value error on each sample is less than a parameter ϵ as shown in Equation 8.

$$\begin{aligned} &\min_{\omega,b,\xi,\xi^*} \frac{1}{2}\omega^T\omega + \sum_i \xi_i + \sum_i \xi_i^* \\ &\text{subject to } \omega^T\phi(x_i) + b - y_i < \epsilon + \xi_i \\ &\qquad y_i - \omega^T\phi(x_i) - b < \epsilon + \xi_i^* \\ &\qquad \xi_i, \xi_i^* > 0 \;\forall\, i \end{aligned} \tag{8}$$

The prediction for a new example x is $f(x) = \omega^T x + b$. LIBSVM was used once again for SVR with a radial basis kernel. The parameter ϵ was set to 0.1.

5.4.3. *Support Vector Machines for Ordinal Regression*

Ordinal regression or ranking learning aims to build a model to predict ordinal ranks. SVOR [28] adds constraints so that the ordering between the scores are accounted for. The problem solved is in Equation 9 where j goes from 1 to $r - 1$ and $C > 0$. The idea is to find an optimal weight vector ω as in SVMClass but also $r - 1$ thresholds b_j defining parallel discriminant hyperplanes for the r labels. In short, for each example x_i^j (the upperscript j means that

[2] Jekabsons G., ARESLab: Adaptive Regression Splines toolbox for Matlab/Octave, 2011, available at http://www.cs.rtu.lv/jekabsons/

[3] http://wwww.gaussianprocesses.org/gpml

	Manhattan error	Euclidean error	Spearman's ρ	Cohen's κ
MLR	0.600 ± 0.007	0.792 ± 0.009	0.831 ± 0.004	0.364 ± 0.008
MARS	0.608 ± 0.008	0.799 ± 0.01	0.821 ± 0.002	0.368 ± 0.009
GP	0.593 ± 0.007	0.853 ± 0.01	0.833 ± 0.005	0.408 ± 0.006
SVR	0.559 ± 0.007	**0.766 ± 0.009**	0.837 ± 0.004	0.414 ± 0.008
SVOR	**0.453 ± 0.005**	**0.763 ± 0.008**	**0.842 ± 0.005**	**0.489 ± 0.006**
SVMClass	0.580 ± 0.009	0.965 ± 0.01	0.776 ± 0.004	0.421 ± 0.005

Table 3. Results of the 10-fold cross validation on the LEGO corpus. 95% confidence interval bounds are provided for each metric.

x_i^j belongs to the j-th category), the function value $\omega^T\phi(x_i^j)$ should be lower than the lower margin $(b_j - 1)$ and each example x_i^{j+1} should have a function value $\omega^T\phi(x_i^{j+1})$ higher than the upper margin $(b_{j+1} - 1)$. More details about this algorithm can be found in [28].

$$\begin{aligned}
&\min_{\omega,b,\xi,\xi^*} \frac{1}{2}\omega^T\omega + C\sum_{j=1}^{r-1}\left(\sum_{i=1}^{n^j}\xi_i^j + \sum_{i=1}^{n^{j+1}}\xi_i^{*j+1}\right)\\
&\text{subject to } \omega^T\phi(x_i^j) - b_j \leq -1 + \xi_i^j\\
&\qquad \xi_i^j \geq 0,\ \text{for } i = 1...n^j\\
&\qquad \omega^T\phi(x_i^{j+1}) - b_j \geq 1 - \xi_i^{*j+1}\\
&\qquad \xi_i^{*j+1} \geq 0,\ \text{for } i = 1...n^{j+1}\\
&\qquad b_{j-1} \leq b_j \text{ for } j = 2, ..., r-1 \qquad (9)
\end{aligned}$$

A new example x is given the label $\text{argmin}_i\{i \mid \omega^T\phi(x) < b_i\}$. We computed SVOR with a Gaussian kernel using the C program available at http://www.gatsby.ucl.ac.uk/chuwei/svor.htm.

6. RESULTS

The results of the test are given in Table 3. For the regression techniques, Cohen's κ coefficient was computed on the rounded predictions made by the models.

Surprisingly, MLR performed quite well on the corpus. Schmitt *et. al* applied MLR to the same corpus and after rounding the returned values, they obtained a κ of 0.35 and a ρ of only 0.46. When we round the values returned by MLR, we obtain a ρ equal to 0.805. This difference of results should be explained by the features sets used in both cases. We also believe splitting categorical features into a set of boolean variables is responsible for this difference. GP performed better than MLR and MARS except concerning the Euclidean error. Since MLR and MARS minimise this error while GP does not, this result could be expected. SVR, on the other hand, performed better than the previously mentioned techniques on all metrics. A significant improvement can be observed in particular concerning the Euclidean and Manhattan errors. The results of SVR are quite good because the constraints on the function value inferred by SVR imposed a very low absolute error on the samples.

SVOR outperforms all these methods on each metric. A significant improvement is done on the Manhattan error and Cohen's κ. The Manhattan error is improved of 0.1 point compared to SVR. Cohen's κ is improved of 0.06 point compared to SVMClass.

SVMClass has the lowest ranking correlation coefficient. This supports the fact that classification techniques are not the most appropriate for IQ prediction because they ignore the constraint imposed by the order of the labels. Nevertheless, as expected SVMClass has the second highest κ coefficient behind SVOR. The classification performances of SVMClass and SVOR can be further compared thanks to their confusion matrices (Table 4). The results predicted by SVOR deviated less from the real values than the ones predicted by SVMClass. For instance, a true 1 was predicted as a 3, a 4 or a 5 88 times with SVOR against 236 times with SVMClass. The recall and precision for critical values 1 and 2 are also significantly higher with SVOR.

SVOR	T 1	T 2	T 3	T 4	T 5	Precision
P 1	572	99	32	4	1	0.808
P 2	124	145	118	33	1	0.344
P 3	69	236	378	228	23	0.405
P 4	18	76	347	716	359	0.472
P 5	1	1	20	269	1410	0.829
Recall	0.730	0.260	0.422	0.573	0.786	
SVMC	T 1	T 2	T 3	T 4	T 5	Precision
P 1	541	95	55	27	3	0.719
P 2	57	70	55	23	2	0.183
P 3	98	188	253	164	19	0.313
P 4	81	182	432	626	274	0.440
P 5	7	22	100	410	1496	0.781
Recall	0.690	0.126	0.282	0.501	0.834	

Table 4. Confusion matrices for SVOR and SVMClass. T i means that i is the true value and P i means i is the predicted value.

7. CONCLUSION

This paper suggested ordinal regression to model interaction quality prediction at a system-user exchange level. Regression and classification models were tested on a set of evaluated dialogues and it was shown that ordinal regression provided the best results for each metric, namely Euclidean and Manhattan errors, Spearman's rank correlation coefficient and Cohen's κ. These good results are explained by the fact that ordinal regression, unlike standard classification or regression, explicitly accounts for the natural ordering of the interaction quality ratings. Future work will consist of using ordinal regression on a corpus of annotated dialogues with a reinforcement-learning based system to infer a reward function from the ratings.

8. REFERENCES

[1] Laila Dybkjaer, Niels O. Bernsen, and Wolfgang Minker, "Evaluation and usability of multimodal spoken language dialogue systems," *Speech Communication*, 2004.

[2] Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella, "PARADISE: a framework for evaluating spoken dialogue agents," in *Proc of EACL*, 1997.

[3] Kate S. Hone and Robert Graham, "Towards a tool for the subjective assessment of speech system interfaces (SASSI)," *Natural Language Engineering*, 2000.

[4] Mikko Hartikainen, EsaPekka Salonen, and Markku Turunen, "Subjective evaluation of spoken dialogue systems using SERVQUAL method," in *Proc of Interspeech*, 2004.

[5] Keelan Evanini, Phillip Hunter, Jackson Liscombe, David Suendermann, Krishna Dayanidhi, and Roberto Pieraccini, "Caller experience: A method for evaluating dialog systems and its automatic prediction," in *Proceedings of IEEE SLT*, 2008.

[6] Alexander Schmitt, Benjamin Schatz, and Wolfgang Minker, "Modeling and predicting quality in spoken human-computer interaction," in *Proc of SIGDIAL*, 2011.

[7] Esther Levin, Roberto Pieraccini, and Wieland Eckert, "Learning dialogue strategies within the markov decision process framework," in *Proc of IEEE ASRU*, 1997.

[8] Olivier Pietquin, Matthieu Geist, Senthilkumar Chandramohan, and Hervé Frezza-Buet, "Sample-efficient batch reinforcement learning for dialogue management optimization," *ACM Transaction on Speech and Language Processing*, 2011.

[9] Lucie Daubigney, Milica Gasic, Senthilkumar Chandramohan, Matthieu Geist, Olivier Pietquin, and Steve Young, "Uncertainty management for on-line optimisation of a pomdp-based large-scale spoken dialogue system," in *Proc of Interspeech*, 2011.

[10] Klaus-Peter Engelbrecht, Florian Gödde, Felix Hartard, Hamed Ketabdar, and Sebastian Möller, "Modeling user satisfaction with hidden markov model," in *Proc of SIGDIAL*, 2009.

[11] Charles Manski, "Regression," *Journal of Economic Literature*, 1991.

[12] Ronald A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, 1936.

[13] Alexander Schmitt, Stefan Ultes, and Wolfgang Minker, "A parameterized and annotated spoken dialog corpus of the cmu let's go bus information system," in *Proc of LREC*, 2012.

[14] Antoine Raux, Brian Langner, Allan Black, and Maxine Eskenazi, "LET'S GO: Improving Spoken Dialog Systems for the Elderly and Non-natives," in *Proc of Eurospeech*, 2003.

[15] Sebastian Möller, *Assessment and Prediction of Speech Quality in Telecommunications*, Kluwer, 2000.

[16] Sebastian Möller, *Quality of telephone-based spoken dialogue systems*, chapter Quality of spoken dialogue systems, Springer, 2005.

[17] Romain Laroche, Ghislain Putois, Philippe Bretier, Martin Aranguren, Julia Velkovska, Helen Hastie, Simon Keizer, Kai Yu, Filip Jurcicek, Oliver Lemon, and Steve Young, "Report D6.4 : Final evaluation of classic towninfo and appointment scheduling systems," Tech. Rep., CLASSIC Project, 2011.

[18] Silke Witt, "A global experience metric for dialog management in spoken dialog systems," in *Proc of SemDial*, 2011.

[19] Satinder Singh, Michael Kearns, Diane Litman, and Marilyn Walker, "Reinforcement learning for spoken dialogue systems," in *Proc of NIPS*, 1999.

[20] Jason D. Williams and Steve Young, "Partially observable markov decision processes for spoken dialog systems," *Computer Speech and Language*, 2007.

[21] Lucie Daubigney, Matthieu Geist, Senthilkumar Chandramohan, and Olivier Pietquin, "A Comprehensive Reinforcement Learning Framework for Dialogue Management Optimisation," *IEEE Journal of Selected Topics in Signal Processing*, 2012.

[22] Marilyn A. Walker, Jeanne C. Fromer, and Shrikanth Narayanan, "Learning optimal dialogue strategies: A case study of a spoken dialogue agent for email," in *Proc of COLING/ACL*, 1998.

[23] Layla El-Asri, Romain Laroche, and Olivier Pietquin, "Reward function learning for dialogue management," in *Proc of STAIRS*, 2012.

[24] Nicole Yankelovitch, "How do users know what to say ?," *Interactions*, 1996.

[25] Diane J. Litman and Shimei Pan, "Designing and evaluating an adaptive spoken dialogue system," *User Modeling and User-Adapted Interaction*, 2002.

[26] Sunao Hara, Norihide Kitaoka, and Kazuya Takeda, "Estimation method of user satisfaction using n-gram-based dialog history model for spoken dialog system," in *Proc of LREC*, 2010.

[27] Amnon Shashua and Anat Levin, "Ranking with large margin principle: Two approaches," in *Proc of NIPS*, 2002.

[28] Wei Chu and S. Sathiya Keerthi, "Support vector ordinal regression," *Neural Computation*, 2007.

[29] Paul Weng, "Ordinal Decision Models for Markov Decision Processes," in *Proc of ECAI*, 2012.

[30] Charles Spearman, "The proof and measurement of association between two things," *American Journal of Psychology*, 1904.

[31] Jacob Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, 1960.

[32] Jerome H. Friedman, "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, 1991.

[33] Carl Edward Rasmussen and Christopher K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, The MIT Press, 2005.

[34] Tobias Jung and Peter Stone, "Feature selection for value function approximation using bayesian model selection," in *Proc of ECML-PKDD*, 2009.

[35] Yaakov Engel, Shie Mannor, and Ron Meir, "Reinforcement learning with gaussian processes," in *Proc of ICML*, 2005.

[36] Vladimir N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons Inc, 1998.

[37] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, 2011.

[38] Harris Drucker, Chris J.C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik, "Support Vector Regression Machines," in *Proceedings of NIPS*, 1996.