Using Machine Learning to Explore Human Multimodal Clarification Strategies

Verena Rieser
Department of Computational Linguistics
Saarland University
Saarbrücken, D-66041
vrieser@coli.uni-sb.de

Oliver Lemon
School of Informatics
University of Edinburgh
Edinburgh, EH8 9LW, GB
olemon@inf.ed.ac.uk

Abstract

We investigate the use of machine learning in combination with feature engineering techniques to explore human multimodal clarification strategies and the use of those strategies for dialogue systems. We learn from data collected in a Wizard-of-Oz study where different wizards could decide whether to ask a clarification request in a multimodal manner or else use speech alone. We show that there is a uniform strategy across wizards which is based on multiple features in the context. These are generic runtime features which can be implemented in dialogue systems. Our prediction models achieve a weighted f-score of 85.3% (which is a 25.5% improvement over a one-rule baseline). To assess the effects of models, feature discretisation, and selection, we also conduct a regression analysis. We then interpret and discuss the use of the learnt strategy for dialogue systems. Throughout the investigation we discuss the issues arising from using small initial Wizard-of-Oz data sets, and we show that feature engineering is an essential step when learning from such limited data.

1 Introduction

Good clarification strategies in dialogue systems help to ensure and maintain mutual understanding and thus play a crucial role in robust conversational interaction. In dialogue application domains with high interpretation uncertainty, for example caused by acoustic uncertainties from a speech recogniser, multimodal generation and input leads to more robust interaction (Oviatt, 2002) and reduced cognitive load (Oviatt et al., 2004). In this paper we investigate the use of machine learning (ML) to explore human multimodal clarification strategies and the use of those strategies to decide, based on the current dialogue context, when a dialogue system's clarification request (CR) should be generated in a multimodal manner.

In previous work (Rieser and Moore, 2005) we showed that for spoken CRs in human-human communication people follow a context-dependent clarification strategy which systematically varies across domains (and even across Germanic languages). In this paper we investigate whether there exists a context-dependent "intuitive" human strategy for multimodal CRs as well. To test this hypothesis we gathered data in a Wizard-of-Oz (WOZ) study, where different wizards could decide when to show a screen output. From this data we build prediction models, using supervised learning techniques together with feature engineering methods, that may explain the underlying process which generated the data. If we can build a model which predicts the data quite reliably, we can show that there is a uniform strategy that the majority of our wizards followed in certain contexts.

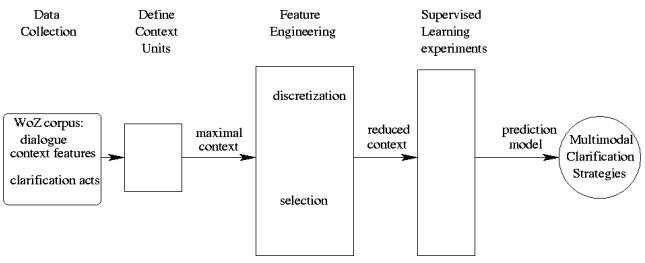

Figure 1: Methodology and structure

The overall method and corresponding structure of the paper is as shown in figure 1. We proceed

as follows. In section 2 we present the WOZ corpus from which we extract a potential context using "Information State Update" (ISU)-based features (Lemon et al., 2005), listed in section 3. We also address the question how to define a suitable "local" context definition for the wizard actions. We apply the feature engineering methods described in section 4 to address the questions of unique thresholds and feature subsets across wizards. These techniques also help to reduce the context representation and thus the feature space used for learning. In section 5 we test different classifiers upon this reduced context and separate out the independent contribution of learning algorithms and feature engineering techniques. In section 6 we discuss and interpret the learnt strategy. Finally we argue for the use of reinforcement learning to optimise the multimodal clarification strategy.

2 The WOZ Corpus

The corpus we are using for learning was collected in a multimodal WOZ study of German task-oriented dialogues for an in-car music player application, (Kruijff-Korbayová et al., 2005) . Using data from a WOZ study, rather than from real system interactions, allows us to investigate how humans clarify. In this study six people played the role of an intelligent interface to an MP3 player and were given access to a database of information. 24 subjects were given a set of predefined tasks to perform using an MP3 player with a multimodal interface. In one part of the session the users also performed a primary driving task, using a driving simulator. The wizards were able to speak freely and display the search results or the playlist on the screen by clicking on various pre-computed templates. The users were also able to speak, as well as make selections on the screen. The user's utterances were immediately transcribed by a typist. The transcribed user's speech was then corrupted by deleting a varying number of words, simulating understanding problems at the acoustic level. This (sometimes) corrupted transcription was then presented to the human wizard. Note that this environment introduces uncertainty on several levels, for example multiple matches in the database, lexical ambiguities, and errors on the acoustic level, as described in (Rieser et al., 2005). Whenever the wizard produced a CR, the experiment leader invoked a questionnaire window on a GUI, where the wizard classified their CR according to the primary source of the understanding problem, mapping to the categories defined by (Traum and Dillenbourg, 1996).

2.1 The Data

The corpus gathered with this setup comprises 70 dialogues, 1772 turns and 17076 words. Example 1 shows a typical multimodal clarification sub-dialogue, [1] concerning an uncertain reference (note that "Venus" is an album name, song title, and an artist name), where the wizard selects a screen output while asking a CR.

(1) **User:** Please play "Venus".
Wizard: Does this list contain the song?
[shows list with 20 DB matches]
User: Yes. It's number 4. *[clicks on item 4]*

For each session we gathered logging information which consists of e.g., the transcriptions of the spoken utterances, the wizard's database query and the number of results, the screen option chosen by the wizard, classification of CRs, etc. We transformed the log-files into an XML structure, consisting of sessions per user, dialogues per task, and turns.[2]

2.2 Data analysis:

Of the 774 wizard turns 19.6% were annotated as CRs, resulting in 152 instances for learning, where our six wizards contributed about equal proportions. A χ^2 test on multimodal strategy (i.e. showing a screen output or not with a CR) showed significant differences between wizards ($\chi^2(1) = 34.21, p < .000$). On the other hand, a Kruskal-Wallis test comparing user preference for the multimodal output showed no significant difference across wizards (H(5)=10.94, $p > .05$). [3] Mean performance ratings for the wizards' multimodal behaviour ranged from 1.67 to 3.5 on a five-point Likert scale. Observing significantly different strategies which are not significantly different in terms of user satisfaction, we conjecture that the wizards converged on strategies which were appropriate in certain *contexts*. To strengthen this

[1] Translated from German.

[2] Where a new "turn" begins at the start of each new user utterance after a wizard utterance, taking the user utterance as a most basic unit of dialogue progression as defined in (Paek and Chickering, 2005).

[3] The Kruskal-Wallis test is the non-parametric equivalent to a one-way ANOVA. Since the users indicated their satisfaction on a 5-point likert scale, an ANOVA which assumes normality would be invalid.

hypothesis we split the data by wizard and and performed a Kruskal-Wallis test on multimodal behaviour per session. Only the two wizards with the lowest performance score showed no significant variation across session, whereas the wizards with the highest scores showed the most varying behaviour. These results again indicate a context dependent strategy. In the following we test this hypothesis (that good multimodal clarification strategies are context-dependent) by building a prediction model of the strategy an *average* wizard took dependent on certain context features.

3 Context/Information-State Features

A state or context in our system is a dialogue information state as defined in (Lemon et al., 2005). We divide the types of information represented in the dialogue information state into *local features* (comprising low level and dialogue features), *dialogue history features*, and *user model features*. We also defined features reflecting the application environment (e.g. `driving`). All features are automatically extracted from the XML log-files (and are available at runtime in ISU-based dialogue systems). From these features we want to learn whether to generate a screen output (`graphic-yes`), or whether to clarify using speech only (`graphic-no`). The case that the wizard only used screen output for clarification did not occur.

3.1 Local Features

First, we extracted features present in the "local" context of a CR, such as the number of matches returned from the data base query (`DBmatches`), how many words were deleted by the corruption algorithm[4] (`deletion`), what problem source the wizard indicated in the pop-up questionnaire (`source`), the previous user speech act (`userSpeechAct`), and the delay between the last wizard utterance and the user's reply (`delay`). [5]

One decision to take for extracting these local features was how to define the "local" context of a CR. As shown in table 1, we experimented with a number of different context definitions. Context 1 defined the local context to be the current turn only, i.e. the turn containing the CR. Context 2 also considered the current turn and the turn following (and is thus not a "runtime" context). Context 3 considered the current turn and the previous turn. Context 4 is the maximal definition of a local context, namely the previous, current, and next turn (also not available at runtime). [6]

id	**Context (turns)**	**acc/ wf-score majority(%)**	**acc/ wf-score Naïve Bayes (%)**
1	only current turn	83.0/54.9	81.0/68.3
2	current and next	71.3/50.4	72.01/68.2
3	**current and previous**	60.50/59.8	76.0*/75.3
4	previous, current, next	67.8/48.9	76.9*/ 74.8

Table 1: Comparison of context definitions for local features (* denotes $p < .05$)

To find the context type which provides the richest information to a classifier, we compared the accuracy achieved in a 10-fold cross validation by a Naïve Bayes classifier (as a standard) on these data sets against the majority class baseline, using a paired t-test, we found that that for context 3 and context 4, Naïve Bayes shows a significant improvement (with $p < .05$ using Bonferroni correction). In table 1 we also show the weighted f-scores since they show that the high accuracy achieved using the first two contexts is due to over-prediction. We chose to use context 3, since these features will be available during system runtime and the learnt strategy could be implemented in an actual system.

3.2 Dialogue History Features

The history features account for events in the whole dialogue so far, i.e. all information gathered before asking the CR, such as the number of CRs asked (`CRhist`), how often the screen output was already used (`screenHist`), the corruption rate so far (`delHist`), the dialogue duration so far (`duration`), and whether the user reacted to the screen output, either by verbally referencing (`refHist`) , e.g. using expressions such as "It's item number 4", or by clicking (`clickHist`) as in example 1.

3.3 User Model Features

Under "user model features" we consider features reflecting the wizards' responsiveness to the be-

[4] Note that this feature is only an approximation of the ASR confidence score that we would expect in an automated dialogue system. See (Rieser et al., 2005) for full details.

[5] We introduced the `delay` feature to handle clarifications concerning contact.

[6] Note that dependent on the context definition a CR might get annotated differently, since placing the question and showing the graphic might be asynchronous events.

haviour and situation of the user. Each session comprised four dialogues with one wizard. The user model features average the user's behaviour in these dialogues so far, such as how responsive the user is towards the screen output, i.e. how often this user clicks (`clickUser`) and how frequently s/he uses verbal references (`refUser`); how often the wizard had already shown a screen output (`screenUser`) and how many CRs were already asked (`CRuser`); how much the user's speech was corrupted on average (`delUser`), i.e. an approximation of how well this user is recognised; and whether this user is currently driving or not (`driving`). This information was available to the wizard.

```
LOCAL FEATURES
        DBmatches: 20
        deletion: 0
        source: reference resolution
        userSpeechAct: command
        delay: 0

HISTORY FEATURES
        [CRhist, screenHist, delHist,
         refHist, clickHist]=0
        duration= 10s

USER MODEL FEATURES
        [clickUser,refUser,screenUser,
         CRuser]=0
        driving= true
```

Figure 2: Features in the context after the first turn in example 1.

3.4 Discussion

Note that all these features are generic over information-seeking dialogues where database results can be displayed on a screen; except for `driving` which only applies to hands-and-eyes-busy situations. Figure 2 shows a context for example 1, assuming that it was the first utterance by this user.

This potential feature space comprises 18 features, many of them taking numeric attributes as values. Considering our limited data set of 152 training instances we run the risk of severe data sparsity. Furthermore we want to explore which features of this potential feature space influenced the wizards' multimodal strategy. In the next two sections we describe feature engineering techniques, namely discretising methods for dimensionality reduction and feature selection methods, which help to reduce the feature space to a subset which is most predictive of multimodal clarification. For our experiments we use implementations of discretisation and feature selection methods provided by the WEKA toolkit (Witten and Frank, 2005).

4 Feature Engineering

4.1 Discretising Numeric Features

Global discretisation methods divide all continuous features into a smaller number of distinct ranges before learning starts. This has two advantages concerning the quality of our data for ML. First, discretisation methods take feature distributions into account and help to avoid sparse data. Second, most of our features are highly positively skewed. Some ML methods (such as the standard extension of the Naïve Bayes classifier to handle numeric features) assume that numeric attributes have a normal distribution. We use Proportional k-Interval (PKI) discretisation as a unsupervised method, and an entropy-based algorithm (Fayyad and Irani, 1993) based on the Minimal Description Length (MDL) principle as a supervised discretisation method.

4.2 Feature Selection

Feature selection refers to the problem of selecting an optimum subset of features that are most predictive of a given outcome. The objective of selection is two-fold: improving the prediction performance of ML models and providing a better understanding of the underlying concepts that generated the data. We chose to apply forward selection for all our experiments given our large feature set, which might include redundant features. We use the following feature filtering methods: correlation-based subset evaluation (CFS) (Hall, 2000) and a decision tree algorithm (rule-based ML) for selecting features before doing the actual learning. We also used a wrapper method called *Selective Naïve Bayes*, which has been shown to perform reliably well in practice (Langley and Sage, 1994). We also apply a correlation-based ranking technique since subset selection models inner-feature relations at the expense of saying less about individual feature performance itself.

4.3 Results for PKI and MDL Discretisation

Feature selection and discretisation influence one-another, i.e. feature selection performs differently on PKI or MDL discretised data. MDL discretisation reduces our range of feature values dramatically. It fails to discretise 10 of 14 numeric features and bars those features from playing a role in the final decision structure because the same discretised value will be given to all instances. However, MDL discretisation cannot replace proper feature selection methods since

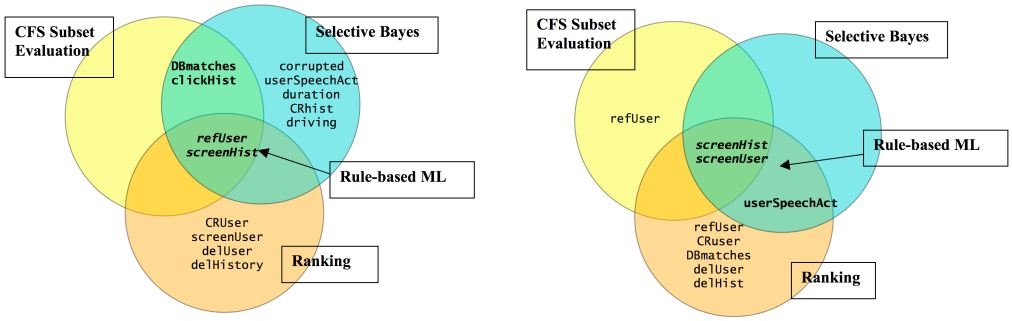

Table 2: Feature selection on PKI-discretised data (left) and on MDL-discretised data (right)

it doesn't explicitly account for redundancy between features, nor for non-numerical features. For the other 4 features which were discretised there is a binary split around one (fairly low) threshold: `screenHist` (.5), `refUser` (.375), `screenUser` (1.0), `CRUser` (1.25).

Table 2 shows two figures illustrating the different subsets of features chosen by the feature selection algorithms on discretised data. From these four subsets we extracted a fifth, using all the features which were chosen by at least two of the feature selection methods, i.e. the features in the overlapping circle regions shown in figure 2. For both data sets the highest ranking features are also the ones contained in the overlapping regions, which are `screenUser`, `refUser` and `screenHist`. For implementation dialogue management needs to keep track of whether the user already saw a screen output in a previous interaction (`screenUser`), or in the same dialogue (`screenHist`), and whether this user (verbally) reacted to the screen output (`refUser`).

5 Performance of Different Learners and Feature Engineering

In this section we evaluate the performance of feature engineering methods in combination with different ML algorithms (where we treat feature optimisation as an integral part of the training process). All experiments are carried out using 10-fold cross-validation. We take an approach similar to (Daelemans et al., 2003) where parameters of the classifier are optimised with respect to feature selection. We use a wide range of different multivariate classifiers which reflect our hypothesis that a decision is based on various features in the context, and compare them against two simple baseline strategies, reflecting deterministic contextual behaviour.

5.1 Baselines

The simplest baseline we can consider is to always predict the majority class in the data, in our case `graphic-no`. This yields a 45.6% wf-score. This baseline reflects a deterministic wizard strategy never showing a screen output.

A more interesting baseline is obtained by using a 1-rule classifier. It chooses the feature which produces the minimum error (which is `refUser` for the PKI discretised data set, and `screenHist` for the MDL set). We use the implementation of a one-rule classifier provided in the WEKA toolkit. This yields a 59.8% wf-score. This baseline reflects a deterministic wizard strategy which is based on a single feature only.

5.2 Machine Learners

For learning we experiment with five different types of supervised classifiers.We chose Naïve Bayes as a joint (generative) probabilistic model, using the WEKA implementation of (John and Langley, 1995)'s classifier; Bayesian Networks as a graphical generative model, again using the WEKA implementation; and we chose maxEnt as a discriminative (conditional) model, using the Maximum Entropy toolkit (Le, 2003). As a rule induction algorithm we used JRIP, the WEKA implementation of (Cohen, 1995)'s Repeated Incremental Pruning to Produce Error Reduction (RIPPER). And for decision trees we used the J4.8 classifier (WEKA's implementation of the C4.5 system (Quinlan, 1993)).

5.3 Comparison of Results

We experimented using these different classifiers on raw data, on MDL and PKI discretised data, and on discretised data using the different feature selection algorithms. To compare the classification outcomes we report on two measures: accuracy and wf-score, which is the weighted

Feature transformation/ (acc./ wf-score (%))	1-rule baseline	Rule Induction	Decision Tree	maxEnt	Naïve Bayes	Bayesian Network	Average
raw data	60.5/**59.8**	76.3/78.3	79.4/78.6	70.0/75.3	76.0/75.3	79.5/72.0	73.62/73.21
PKI + all features	60.5/ 64.6	67.1/66.4	77.4/76.3	70.7/76.7	77.5/81.6	77.3/82.3	71.75/74.65
PKI+ CFS subset	60.5/64.4	68.7/70.7	79.2/76.9	76.7/79.4	78.2/80.6	77.4/80.7	73.45/75.45
PKI+ rule-based ML	60.5/66.5	72.8/76.1	76.0/73.9	75.3/80.2	80.1/78.3	80.8/79.8	74.25/75.80
PKI+ selective Bayes	60.5/64.4	68.2/65.2	78.4/77.9	79.3/78.1	84.6/**85.3**	84.5/84.6	75.92/75.92
PKI+ subset overlap	60.5/64.4	70.9/70.7	75.9/76.9	76.7/78.2	84.0/80.6	83.7/80.7	75.28/75.25
MDL + all features	60.5/69.9	79.0/78.8	78.0/78.1	71.3/76.8	74.9/73.3	74.7/73.9	73.07/75.13
MDL + CFS subset	60.5/69.9	80.1/78.2	80.6/78.2	76.0/80.2	75.7/75.8	75.7/75.8	74.77/76.35
MDL + rule-based ML	60.5/75.5	80.4/81.6	78.7/80.2	79.3/78.8	82.7/82.9	82.7/82.9	77.38/80.32
MDL + select. Bayes	60.5/75.5	80.4/81.6	78.7/80.8	79.3/80.1	82.7/82.9	82.7/82.9	77.38/80.63
MDL + overlap	60.5/75.5	80.4/81.6	78.7/80.8	79.3/80.1	82.7/82.9	82.7/82.9	77.38/80.63
average	60.5/68.24	74.9/75.38	78.26/78.06	75.27/78.54	79.91/79.96	80.16/79.86	

Table 3: Average accuracy and wf-scores for models in feature engineering experiments .

sum (by class frequency in the data; 39.5% `graphic-yes`, 60.5% `graphic-no`) of the f-scores of the individual classes. In table 3 we see fairly stable high performance for Bayesian models with MDL feature selection. However, the best performing model is Naïve Bayes using wrapper methods (selective Bayes) for feature selection and PKI discretisation. This model achieves a wf-score of 85.3%, which is a 25.5% improvement over the 1-rule baseline.

We separately explore the models and feature engineering techniques and their impact on the prediction accuracy for each trial/cross-validation. In the following we separate out the independent contribution of models and features. To assess the effects of models, feature discretisation and selection on performance accuracy, we conduct a hierarchical regression analysis. The models alone explain 18.1% of the variation in accuracy ($R_2 = .181$) whereas discretisation methods only contribute 0.4% and feature selection 1% ($R_2 = .195$). All parameters, except for discretisation methods have a significant impact on modelling accuracy ($P < .001$), indicating that feature selection is an essential step for predicting wizard behaviour. The coefficients of the regression model lead us to the following hypotheses which we explore by comparing the group means for models, discretisation, and features selection methods. Applying a Kruskal-Wallis test with Mann-Whitney tests as a post-hoc procedure (using Bonferroni correction for multiple comparisons), we obtained the following results: [7]

- All ML algorithms are significantly better than the majority and one-rule baselines. All except maxEnt are significantly better than the Rule Induction algorithm. There is no significant difference in the performance of Decision Tree, maxEnt, Naïve Bayes, and Bayesian Network classifiers. Multivariate models being significantly better than the two baseline models indicates that we have a strategy that is based on context features.

- For discretisation methods we found that the classifiers were performing significantly better on MDL discretised data than on PKI or continuous data. MDL being significantly better than continuous data indicates that all wizards behaved as though using thresholds to make their decisions, and MDL being better than PKI supports the hypothesis that decisions were context dependent.

- All feature selection methods (except for CFS) lead to better performance than using all of the features. Selective Bayes and rule-based ML selection performed significantly better than CFS. Selective Bayes, rule-based ML, and subset-overlap showed no significant differences. These results show that wizards behaved as though specific features were important (but they suggest that inner-feature relations used by CFS are less important).

Discussion of results: These experimental results show two things. First, the results indicate that we can learn a good prediction model from our data. We conclude that our six wizards did not behave arbitrarily, but selected their strategy according to certain contextual features. By separating out the individual contributions of models and feature engineering techniques, we have shown that wizard behaviour is based on multiple features. In sum, Decision Tree, max-

[7] We cannot report full details here. Supplementary material is available at `www.coli.uni-saarland.de/~vrieser/acl06-supplementary.html`

Ent, Naïve Bayes, and Bayesian Network classifiers on MDL discretised data using Selective Bayes and Rule-based ML selection achieved the best results. The best performing feature subset was `screenUser,screenHist,` and `userSpeechAct`. The best performing model uses the richest feature space including the feature `driving`.

Second, the regression analysis shows that using these feature engineering techniques in combination with improved ML algorithms is an essential step for learning good prediction models from the small data sets which are typically available from multimodal WOZ studies.

6 Interpretation of the learnt Strategy

For interpreting the learnt strategies we discuss Rule Induction and Decision Trees since they are the easiest to interpret (and to implement in standard rule-based dialogue systems). For both we explain the results obtained by MDL and selective Bayes, since this combination leads to the best performance.

Rule induction: Figure 3 shows a reformulation of the rules from which the learned classifier is constructed. The feature `screenUser` plays a central role. These rules (in combination with the low thresholds) say that if you have already shown a screen output to this particular user in any previous turn (i.e. `screenUser > 1`), then do so again if the previous user speech act was a command (i.e. `userSpeechAct=command`) or if you have already shown a screen output in a previous turn in this dialogue (i.e. `screenHist>0.5`). Otherwise don't show screen output when asking a clarification.

Decision tree: Figure 4 shows the decision tree learnt by the classifier J4.8. The five rules contained in this tree also heavily rely on the user model as well as the previous screen history. The rules constructed by the first two nodes (`screenUser, screenHist`) may lead to a repetitive strategy since the right branch will result in the same action (`graphic-yes`) in all future actions. The only variation is introduced by the speech act, collapsing the tree to the same rule set as in figure 3. Note that this rule-set is based on domain independent features.

Discussion: Examining the classifications made by our best performing Bayesian models we found that the learnt conditional probability distributions produce similar feature-value mappings to the rules described above. The strategy learnt by the classifiers heavily depends on features obtained in previous interactions, i.e. user model features. Furthermore these strategies can lead to repetitive action, i.e. if a screen output was once shown to this user, and the user has previously used or referred to the screen, the screen will be used over and over again.

For learning a strategy which varies in context but adapts in more subtle ways (e.g. to the user model), we would need to explore many more strategies through interactions with users to find an optimal one. One way to reduce costs for building such an optimised strategy is to apply Reinforcement Learning (RL) with simulated users. In future work we will begin with the strategy learnt by supervised learning (which reflects sub-optimal average wizard behaviour) and optimise it for different user models and reward structures.

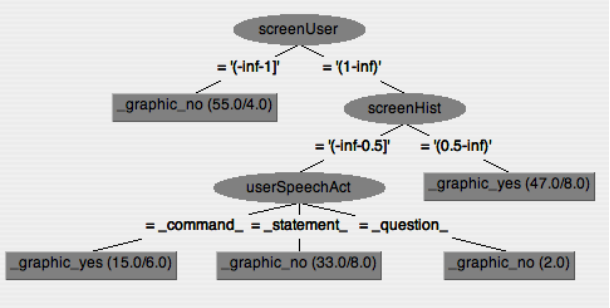

Figure 4: Five-rule tree from J4.8 ("inf" = ∞)

7 Summary and Future Work

We showed that humans use a context-dependent strategy for asking multimodal clarification requests by learning such a strategy from WOZ data. Only the two wizards with the lowest performance scores showed no significant variation across sessions, leading us to hypothesise that the better wizards converged on a context-dependent strategy. We were able to discover a runtime context based on which all wizards behaved uniformly, using feature discretisation methods and feature selection methods on dialogue context features. Based on these features we were able to predict how an 'average' wizard would behave in that context with an accuracy of 84.6% (wf-score of 85.3%, which is a 25.5% improvement over a one rule-based baseline). We explained the learned strategies and showed that they can be implemented in

```
IF screenUser>1 AND (userSpeechAct=command OR screenHist>0.5) THEN graphic=yes
ELSE graphic=no
```

Figure 3: Reformulation of the rules learnt by JRIP

rule-based dialogue systems based on domain independent features. We also showed that feature engineering is essential for achieving significant performance gains when using large feature spaces with the small data sets which are typical of dialogue WOZ studies. By interpreting the learnt strategies we found them to be sub-optimal. In current research, RL is applied to optimise strategies and has been shown to lead to dialogue strategies which are better than those present in the original data (Henderson et al., 2005). The next step towards a RL-based system is to add task-level and reward-level annotations to calculate reward functions, as discussed in (Rieser et al., 2005). We furthermore aim to learn more refined clarification strategies indicating the problem source and its severity.

Acknowledgements

The authors would like thank the ACL reviewers, Alissa Melinger, and Joel Tetreault for help and discussion. This work is supported by the TALK project, `www.talk-project.org`, and the International Post-Graduate College for Language Technology and Cognitive Systems, Saarbrücken.

References

William W. Cohen. 1995. Fast effective rule induction. In *Proceedings of the 12th ICML-95*.

Walter Daelemans, Véronique Hoste, Fien De Meulder, and Bart Naudts. 2003. Combined optimization of feature selection and algorithm parameter interaction in machine learning of language. In *Proceedings of the 14th ECML-03*.

Usama Fayyad and Keki Irani. 1993. Multi-interval discretization of continuousvalued attributes for classification learning. In *Proc. IJCAI-93*.

Mark Hall. 2000. Correlation-based feature selection for discrete and numeric class machine learning. In *Proc. 17th Int Conf. on Machine Learning*.

James Henderson, Oliver Lemon, and Kallirroi Georgila. 2005. Hybrid Reinforcement/Supervised Learning for Dialogue Policies from COMMUNICATOR data. In *IJCAI workshop on Knowledge and Reasoning in Practical Dialogue Systems,*.

George John and Pat Langley. 1995. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the 11th UAI-95*. Morgan Kaufmann.

Ivana Kruijff-Korbayová, Nate Blaylock, Ciprian Gerstenberger, Verena Rieser, Tilman Becker, Michael Kaisser, Peter Poller, and Jan Schehl. 2005. An experiment setup for collecting data for adaptive output planning in a multimodal dialogue system. In *10th European Workshop on NLG*.

Pat Langley and Stephanie Sage. 1994. Induction of selective bayesian classifiers. In *Proceedings of the 10th UAI-94*.

Zhang Le. 2003. Maximum entropy modeling toolkit for Python and C++.

Oliver Lemon, Kallirroi Georgila, James Henderson, Malte Gabsdil, Ivan Meza-Ruiz, and Steve Young. 2005. Deliverable d4.1: Integration of learning and adaptivity with the ISU approach.

Sharon Oviatt, Rachel Coulston, and Rebecca Lunsford. 2004. When do we interact multimodally? Cognitive load and multimodal communication patterns. In *Proceedings of the 6th ICMI-04*.

Sharon Oviatt. 2002. Breaking the robustness barrier: Recent progress on the design of robust multimodal systems. In *Advances in Computers*. Academic Press.

Tim Paek and David Maxwell Chickering. 2005. The markov assumption in spoken dialogue management. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*.

Ross Quinlan. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann.

Verena Rieser and Johanna Moore. 2005. Implications for Generating Clarification Requests in Task-oriented Dialogues. In *Proceedings of the 43rd ACL*.

Verena Rieser, Ivana Kruijff-Korbayová, and Oliver Lemon. 2005. A corpus collection and annotation framework for learning multimodal clarification strategies. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*.

David Traum and Pierre Dillenbourg. 1996. Miscommunication in multi-modal collaboration. In *Proceedings of the Workshop on Detecting, Repairing, and Preventing Human-Machine Miscommunication*. AAAI-96.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*. Morgan Kaufmann.