Natural Language Generation as Planning Under Uncertainty for Spoken Dialogue Systems

Verena Rieser
School of Informatics
University of Edinburgh
vrieser@inf.ed.ac.uk

Oliver Lemon
School of Informatics
University of Edinburgh
olemon@inf.ed.ac.uk

Abstract

We present and evaluate a new model for Natural Language Generation (NLG) in Spoken Dialogue Systems, based on statistical planning, given noisy feedback from the current generation context (e.g. a user and a surface realiser). We study its use in a standard NLG problem: how to present information (in this case a set of search results) to users, given the complex trade-offs between utterance length, amount of information conveyed, and cognitive load. We set these trade-offs by analysing existing MATCH data. We then train a NLG policy using Reinforcement Learning (RL), which adapts its behaviour to noisy feedback from the current generation context. This policy is compared to several baselines derived from previous work in this area. The learned policy significantly outperforms all the prior approaches.

1 Introduction

Natural language allows us to achieve the same communicative goal ("what to say") using many different expressions ("how to say it"). In a Spoken Dialogue System (SDS), an abstract communicative goal (CG) can be generated in many different ways. For example, the CG to present database results to the user can be realized as a summary (Polifroni and Walker, 2008; Demberg and Moore, 2006), or by comparing items (Walker et al., 2004), or by picking one item and recommending it to the user (Young et al., 2007).

Previous work has shown that it is useful to adapt the generated output to certain features of the dialogue context, for example user preferences, e.g. (Walker et al., 2004; Demberg and Moore, 2006), user knowledge, e.g. (Janarthanam and Lemon, 2008), or predicted TTS quality, e.g. (Nakatsu and White, 2006).

In extending this previous work we treat NLG as a statistical sequential planning problem, analogously to current statistical approaches to Dialogue Management (DM), e.g. (Singh et al., 2002; Henderson et al., 2008; Rieser and Lemon, 2008a) and "conversation as action under uncertainty" (Paek and Horvitz, 2000). In NLG we have similar trade-offs and unpredictability as in DM, and in some systems the content planning and DM tasks are overlapping. Clearly, very long system utterances with many actions in them are to be avoided, because users may become confused or impatient, but each individual NLG action will convey some (potentially) useful information to the user. There is therefore an optimization problem to be solved. Moreover, the user judgements or next (most likely) action after each NLG action are unpredictable, and the behaviour of the surface realizer may also be variable (see Section 6.2).

NLG could therefore fruitfully be approached as a sequential statistical planning task, where there are trade-offs and decisions to make, such as whether to choose another NLG action (and which one to choose) or to instead stop generating. Reinforcement Learning (RL) allows us to optimize such trade-offs in the presence of uncertainty, i.e. the chances of achieving a better state, while engaging in the risk of choosing another action.

In this paper we present and evaluate a new model for NLG in Spoken Dialogue Systems as planning under uncertainty. In Section 2 we argue for applying RL to NLG problems and explain the overall framework. In Section 3 we discuss challenges for NLG for Information Presentation. In Section 4 we present results from our analysis of the MATCH corpus (Walker et al., 2004). In Section 5 we present a detailed example of our proposed NLG method. In Section 6 we report on experimental results using this framework for exploring Information Presentation policies. In Section 7 we conclude and discuss future directions.

Proceedings of the 12th Conference of the European Chapter of the ACL, pages 683–691,
Athens, Greece, 30 March – 3 April 2009.

2 NLG as planning under uncertainty

We adopt the general framework of NLG as planning under uncertainty (see (Lemon, 2008) for the initial version of this approach). Some aspects of NLG have been treated as planning, e.g. (Koller and Stone, 2007; Koller and Petrick, 2008), but never before as statistical planning.

NLG actions take place in a stochastic environment, for example consisting of a user, a realizer, and a TTS system, where the individual NLG actions have uncertain effects on the environment. For example, presenting differing numbers of attributes to the user, and making the user more or less likely to choose an item, as shown by (Rieser and Lemon, 2008b) for multimodal interaction.

Most SDS employ fixed template-based generation. Our goal, however, is to employ a stochastic realizer for SDS, see for example (Stent et al., 2004). This will introduce additional noise, which higher level NLG decisions will need to react to. In our framework, the NLG component must achieve a high-level Communicative Goal from the Dialogue Manager (e.g. to present a number of items) through planning a sequence of lower-level generation steps or actions, for example first to summarize all the items and then to recommend the highest ranking one. Each such action has unpredictable effects due to the stochastic realizer. For example the realizer might employ 6 attributes when recommending item i_4, but it might use only 2 (e.g. price and cuisine for restaurants), depending on its own processing constraints (see e.g. the realizer used to collect the MATCH project data). Likewise, the user may be likely to choose an item after hearing a summary, or they may wish to hear more. Generating appropriate language in context (e.g. attributes presented so far) thus has the following important features in general:

- NLG is *goal driven* behaviour
- NLG must plan a *sequence* of actions
- each action *changes* the environment state or context
- the effect of each action is *uncertain*.

These facts make it clear that the problem of planning how to generate an utterance falls naturally into the class of statistical planning problems, rather than rule-based approaches such as (Moore et al., 2004; Walker et al., 2004), or supervised learning as explored in previous work, such as classifier learning and re-ranking, e.g. (Stent et al., 2004; Oh and Rudnicky, 2002). Supervised approaches involve the ranking of a set of completed plans/utterances and as such cannot adapt online to the context or the user. Reinforcement Learning (RL) provides a principled, data-driven optimisation framework for our type of planning problem (Sutton and Barto, 1998).

3 The Information Presentation Problem

We will tackle the well-studied problem of Information Presentation in NLG to show the benefits of this approach. The task here is to find the best way to present a set of search results to a user (e.g. some restaurants meeting a certain set of constraints). This is a task common to much prior work in NLG, e.g. (Walker et al., 2004; Demberg and Moore, 2006; Polifroni and Walker, 2008).

In this problem, there there are many decisions available for exploration. For instance, which presentation strategy to apply (*NLG strategy selection*), how many attributes of each item to present (*attribute selection*), how to rank the items and attributes according to different models of user preferences (*attribute ordering*), how many (specific) items to tell them about (*conciseness*), how many sentences to use when doing so (*syntactic planning*), and which words to use (*lexical choice*) etc. All these parameters (and potentially many more) can be varied, and ideally, jointly optimised based on user judgements.

We had two corpora available to study some of the regions of this decision space. We utilise the MATCH corpus (Walker et al., 2004) to extract an evaluation function (also known as "reward function") for RL. Furthermore, we utilise the SPaRKy corpus (Stent et al., 2004) to build a high quality stochastic realizer. Both corpora contain data from "overhearer" experiments targeted to Information Presentation in dialogues in the restaurant domain. While we are ultimately interested in how hearers *engaged* in dialogues judge different Information Presentations, results from overhearers are still directly relevant to the task.

4 MATCH corpus analysis

The MATCH project made two data sets available, see (Stent et al., 2002) and (Whittaker et al., 2003), which we combine to define an evaluation function for different Information Presentation strategies.

strategy	example	$av.\#attr$	$av.\#sentence$
SUMMARY	*"The 4 restaurants differ in food quality, and cost."* ($\#attr = 2, \#sentence = 1$)	2.07±.63	1.56±.5
COMPARE	*"Among the selected restaurants, the following offer exceptional overall value. Aureole's price is 71 dollars. It has superb food quality, superb service and superb decor. Daniel's price is 82 dollars. It has superb food quality, superb service and superb decor."* ($\#attr = 4, \#sentence = 5$)	3.2±1.5	5.5±3.11
RECOMMEND	*"Le Madeleine has the best overall value among the selected restaurants. Le Madeleine's price is 40 dollars and It has very good food quality. It's in Midtown West. "* ($\#attr = 3, \#sentence = 3$)	2.4±.7	3.5±.53

Table 1: NLG strategies present in the MATCH corpus with average no. attributes and sentences as found in the data.

The first data set, see (Stent et al., 2002), comprises 1024 ratings by 16 subjects (where we only use the speech-based half, $n = 512$) on the following presentation strategies: RECOMMEND, COMPARE, SUMMARY. These strategies are realized using templates as in Table 2, and varying numbers of attributes. In this study the users rate the individual presentation strategies as significantly different ($F(2) = 1361, p < .001$). We find that SUMMARY is rated significantly worse ($p = .05$ with Bonferroni correction) than RECOMMEND and COMPARE, which are rated as equally good.

This suggests that one should never generate a SUMMARY. However, SUMMARY has different qualities from COMPARE and RECOMMEND, as it gives users a general overview of the domain, and probably helps the user to feel more confident when choosing an item, especially when they are unfamiliar with the domain, as shown by (Polifroni and Walker, 2008).

In order to further describe the strategies, we extracted different surface features as present in the data (e.g. number of attributes realised, number of sentences, number of words, number of database items talked about, etc.) and performed a stepwise linear regression to find the features which were important to the overhearers (following the PARADISE framework (Walker et al., 2000)). We discovered a trade-off between the *length* of the utterance ($\#sentence$) and the number of attributes realised ($\#attr$), i.e. its *informativeness*, where overhearers like to hear as many attributes as possible in the most concise way, as indicated by the regression model shown in Equation 1 ($R^2 = .34$). [1]

$$score = .775 \times \#attr + (-.301) \times \#sentence; \quad (1)$$

The second MATCH data set, see (Whittaker et al., 2003), comprises 1224 ratings by 17 subjects on the NLG strategies RECOMMEND and COMPARE. The strategies realise varying numbers of attributes according to different "conciseness" values: `concise` (1 or 2 attributes), `average` (3 or 4), and `verbose` (4,5, or 6). Overhearers rate all conciseness levels as significantly different ($F(2) = 198.3, p < .001$), with `verbose` rated highest and `concise` rated lowest, supporting our findings in the first data set. However, the relation between number of attributes and user ratings is not strictly linear: ratings drop for $\#attr = 6$. This suggests that there is an upper limit on how many attributes users like to hear. We expect this to be especially true for real users engaged in actual dialogue interaction, see (Winterboer et al., 2007). We therefore include "cognitive load" as a variable when training the policy (see Section 6).

In addition to the trade-off between *length* and *informativeness* for single NLG strategies, we are interested whether this trade-off will also hold for generating *sequences* of NLG actions. (Whittaker et al., 2002), for example, generate a *combined strategy* where first a SUMMARY is used to describe the retrieved subset and then they RECOMMEND one specific item/restaurant. For example *"The 4 restaurants are all French, but differ in*

[1] For comparison: (Walker et al., 2000) report on R^2 between .4 and .5 on a slightly lager data set.

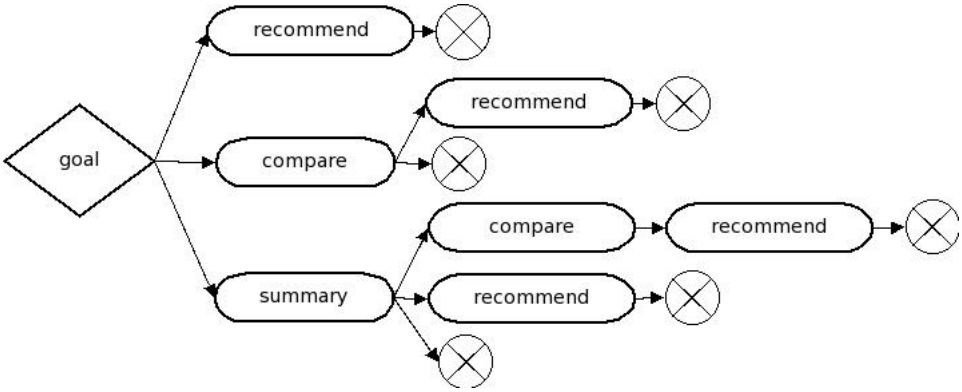

Figure 1: Possible NLG policies (X=stop generation)

food quality, and cost. Le Madeleine has the best overall value among the selected restaurants. Le Madeleine's price is 40 dollars and It has very good food quality. It's in Midtown West."

We therefore extend the set of possible strategies present in the data for exploration: we allow ordered combinations of the strategies, assuming that only COMPARE or RECOMMEND can follow a SUMMARY, and that only RECOMMEND can follow COMPARE, resulting in 7 possible actions:

1. RECOMMEND
2. COMPARE
3. SUMMARY
4. COMPARE+RECOMMEND
5. SUMMARY+RECOMMEND
6. SUMMARY+COMPARE
7. SUMMARY+COMPARE+RECOMMEND

We then analytically solved the regression model in Equation 1 for the 7 possible strategies using average values from the MATCH data. This is solved by a system of linear inequalities. According to this model, the best ranking strategy is to do all the presentation strategies in one sequence, i.e. SUMMARY+COMPARE+RECOMMEND. However, this analytic solution assumes a "one-shot" generation strategy where there is no intermediate feedback from the environment: users are simply static overhearers (they cannot "barge-in" for example), there is no variation in the behaviour of the surface realizer, i.e. one would use fixed templates as in MATCH, and the user has unlimited cognitive capabilities. These assumptions are not realistic, and must be relaxed. In the next Section we describe a worked through example of the overall framework.

5 Method: the RL-NLG model

For the reasons discussed above, we treat the NLG module as a statistical planner, operating in a stochastic environment, and optimise it using Reinforcement Learning. The input to the module is a Communicative Goal supplied by the Dialogue Manager. The CG consists of a Dialogue Act to be generated, for example `present_items`(i_1, i_2, i_5, i_8), and a System Goal (`SysGoal`) which is the desired user reaction, e.g. to make the user choose one of the presented items (`user_choose_one_of`(i_1, i_2, i_5, i_8)). The RL-NLG module must plan a sequence of lower-level NLG actions that achieve the goal (at lowest cost) in the current context. The context consists of a user (who may remain silent, supply more constraints, choose an item, or quit), and variation from the sentence realizer described above.

Now let us walk-through one simple utterance plan as carried out by this model, as shown in Table 2. Here, we start with the CG `present_items`(i_1, i_2, i_5, i_8) & `user_choose_one_of`(i_1, i_2, i_5, i_8) from the system's DM. This initialises the NLG state ($init$). The policy chooses the action SUMMARY and this transitions us to state $s1$, where we observe that 4 attributes and 1 sentence have been generated, and the user is predicted to remain silent. In this state, the current NLG policy is to RECOMMEND the top ranked item (i_5, for this user), which takes us to state $s2$, where 8 attributes have been generated in a total of 4 sentences, and the user chooses an item. The policy holds that in states like $s2$ the

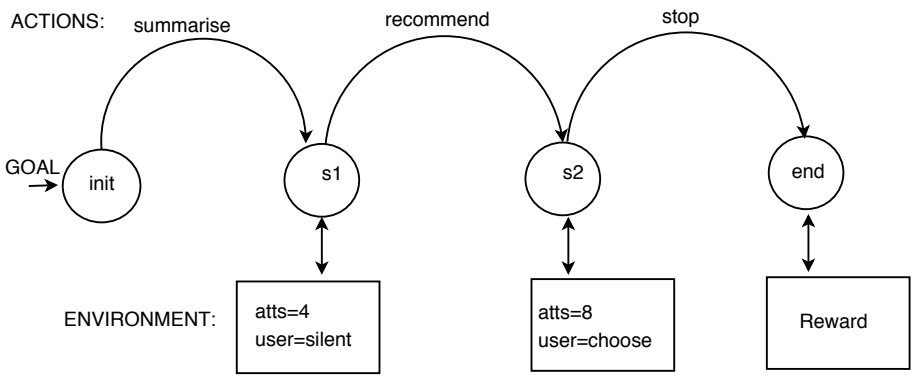

Figure 2: Example RL-NLG action sequence for Table 4

State	Action	State change/effect
init	SysGoal: present_items(i_1, i_2, i_5, i_8) & user_choose_one_of(i_1, i_2, i_5, i_8)	initialise state
s1	RL-NLG: SUMMARY(i_1, i_2, i_5, i_8)	att=4, sent=1, user=silent
s2	RL-NLG: RECOMMEND(i_5)	att=8, sent=4, user=choose(i_5)
end	RL-NLG: stop	calculate Reward

Table 2: Example utterance planning sequence for Figure 2

best thing to do is "stop" and pass the turn to the user. This takes us to the state *end*, where the total reward of this action sequence is computed (see Section 6.3), and used to update the NLG policy in each of the visited state-action pairs via back-propagation.

6 Experiments

We now report on a proof-of-concept study where we train our policy in a simulated learning environment based on the results from the MATCH corpus analysis in Section 4. Simulation-based RL allows to explore unseen actions which are not in the data, and thus less initial data is needed (Rieser and Lemon, 2008b). Note, that we cannot directly learn from the MATCH data, as therefore we would need data from an interactive dialogue. We are currently collecting such data in a Wizard-of-Oz experiment.

6.1 User simulation

User simulations are commonly used to train strategies for Dialogue Management, see for example (Young et al., 2007). A user simulation for NLG is very similar, in that it is a predictive model of the most likely next user act. However, this user act does not actually change the overall dialogue state (e.g. by filling slots) but it only changes the generator state. In other words, the NLG user simulation tells us what the user is most likely to do next, *if we were to stop generating now*. It also tells us the probability whether the user chooses to "barge-in" after a system NLG action (by either choosing an item or providing more information).

The user simulation for this study is a simple bi-gram model, which relates the number of attributes presented to the next likely user actions, see Table 3. The user can either follow the goal provided by the DM (`SysGoal`), for example choosing an item. The user can also do something else (`userElse`), e.g. providing another constraint, or the user can quit (`userQuit`).

For simplification, we discretise the number of attributes into `concise-average-verbose`, reflecting the conciseness values from the MATCH data, as described in Section 4. In addition, we assume that the user's cognitive abilities are limited ("cognitive load"), based on the results from the second MATCH data set in Section 4. Once the number of attributes is more than the "magic number 7" (reflecting psychological results on short-term memory) (Baddeley, 2001)) the user is more likely to become confused and quit.

The probabilities in Table 3 are currently manually set heuristics. We are currently conducting a Wizard-of-Oz study in order to learn these proba-

bilities (and other user parameters) from real data.

	SysGoal	userElse	userQuit
concise	20.0	60.0	20.0
average	60.0	20.0	20.0
verbose	20.0	20.0	60.0

Table 3: NLG bi-gram user simulation

6.2 Realizer model

The sequential NLG model assumes a realizer, which updates the context after each generation step (i.e. after each single action). We estimate the realiser's parameters from the mean values we found in the MATCH data (see Table 1). For this study we first (randomly) vary the number of attributes, whereas the number of sentences is fixed (see Table 4). In current work we replace the realizer model with an implemented generator that replicates the variation found in the SPaRKy realizer (Stent et al., 2004).

	$\#attr$	$\#sentence$
SUMMARY	1 or 2	2
COMPARE	3 or 4	6
RECOMMEND	2 or 3	3

Table 4: Realizer parameters

6.3 Reward function

The reward function defines the final goal of the utterance generation sequence. In this experiment the reward is a function of the various data-driven trade-offs as identified in the data analysis in Section 4: utterance length and number of provided attributes, as weighted by the regression model in Equation 1, as well as the next predicted user action. Since we currently only have overhearer data, we manually estimate the reward for the next most likely user act, to supplement the data-driven model. If in the *end* state the next most likely user act is `userQuit`, the learner gets a penalty of -100, `userElse` receives 0 reward, and `SysGoal` gains $+100$ reward. Again, these hand coded scores need to be refined by a more targeted data collection, but the other components of the reward function are data-driven.

Note that RL learns to "make compromises" with respect to the different trade-offs. For example, the user is less likely to choose an item if there are more than 7 attributes, but the realizer can generate 9 attributes. However, in some contexts it might be desirable to generate all 9 attributes, e.g. if the generated utterance is short. Threshold-based approaches, in contrast, cannot (easily) reason with respect to the current content.

6.4 State and Action Space

We now formulate the problem as a Markov Decision Process (MDP), relating states to actions. Each state-action pair is associated with a *transition probability*, which is the probability of moving from state s at time t to state s' at time $t+1$ after having performed action a when in state s. This transition probability is computed by the environment model (i.e. user and realizer), and explicitly captures noise/uncertainty in the environment. This is a major difference to other non-statistical planning approaches. Each transition is also associated with a reinforcement signal (or reward) r_{t+1} describing how good the result of action a was when performed in state s.

The state space comprises 9 binary features representing the number of attributes, 2 binary features representing the predicted user's next action to follow the system goal or quit, as well as a discrete feature reflecting the number of sentences generated so far, as shown in Figure 3. This results in $2^{11} \times 6 = 12,288$ distinct generation states. We trained the policy using the well known SARSA algorithm, using linear function approximation (Sutton and Barto, 1998). The policy was trained for 3600 simulated NLG sequences.

In future work we plan to learn lower level decisions, such as lexical adaptation based on the vocabulary used by the user.

6.5 Baselines

We derive the baseline policies from Information Presentation strategies as deployed by current dialogue systems. In total we utilise 7 different baselines (B1-B7), which correspond to single branches in our policy space (see Figure 1):

B1: RECOMMEND only, e.g. (Young et al., 2007)

B2: COMPARE only, e.g. (Henderson et al., 2008)

B3: SUMMARY only, e.g. (Polifroni and Walker, 2008)

B4: SUMMARY followed by RECOMMEND, e.g. (Whittaker et al., 2002)

B5: Randomly choosing between COMPARE and RECOMMEND, e.g. (Walker et al., 2007)

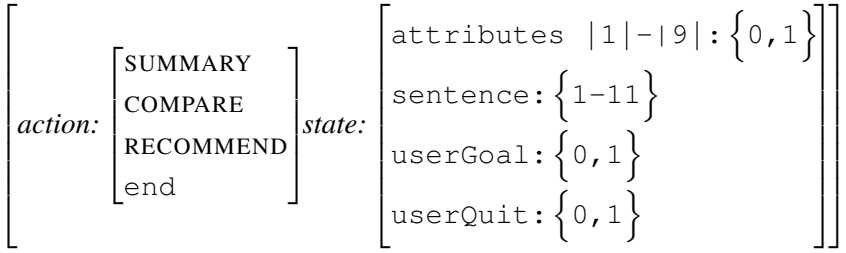

Figure 3: State-Action space for RL-NLG

B6: Randomly choosing between all 7 outputs

B7: Always generating whole sequence, i.e. SUMMARY+COMPARE+RECOMMEND, as suggested by the analytic solution (see Section 4).

6.6 Results

We analyse the test runs (n=200) using an ANOVA with a PostHoc T-Test (with Bonferroni correction). RL significantly ($p < .001$) outperforms all baselines in terms of final reward, see Table 5. RL is the only policy which significantly improves the next most likely user action by adapting to features in the current context. In contrast to conventional approaches, RL learns to 'control' its environment according to the estimated transition probabilities and the associated rewards.

The learnt policy can be described as follows: It either starts with SUMMARY or COMPARE after the *init* state, i.e. it learnt to never start with a RECOMMEND. It stops generating after COMPARE if the `userGoal` is (probably) reached (e.g. the user is most likely to choose an item in the next turn, which depends on the number of attributes generated), otherwise it goes on and generates a RECOMMEND. If it starts with SUMMARY, it always generates a COMPARE afterwards. Again, it stops if the `userGoal` is (probably) reached, otherwise it generates the full sequence (which corresponds to the analytic solution B7).

The analytic solution B7 performs second best, and significantly outperforms all the other baselines ($p < .01$). Still, it is significantly worse ($p < .001$) than the learnt policy as this 'one-shot-strategy' cannot robustly and dynamically adopt to noise or changes in the environment.

In general, generating sequences of NLG actions rates higher than generating single actions only: B4 and B6 rate directly after RL and B7, while B1, B2, B3, B5 are all equally bad given our data-driven definition of reward and environment. Furthermore, the simulated environment allows us to replicate the results in the MATCH corpus (see Section 4) when only comparing single strategies: SUMMARY performs significantly worse, while RECOMMEND and COMPARE perform equally well.

policy	reward	($\pm std$)
B1	99.1	($\pm$129.6)
B2	90.9	($\pm$142.2)
B3	65.5	($\pm$137.3)
B4	176.0	($\pm$154.1)
B5	95.9	($\pm$144.9)
B6	168.8	($\pm$165.3)
B7	229.3	($\pm$157.1)
RL	310.8	($\pm$136.1)

Table 5: Evaluation Results ($p < .001$)

7 Conclusion

We presented and evaluated a new model for Natural Language Generation (NLG) in Spoken Dialogue Systems, based on statistical planning. After motivating and presenting the model, we studied its use in Information Presentation.

We derived a data-driven model predicting users' judgements to different information presentation actions (reward function), via a regression analysis on MATCH data. We used this regression model to set weights in a reward function for Reinforcement Learning, and so optimize a context-adaptive presentation policy. The learnt policy was compared to several baselines derived from previous work in this area, where the learnt policy significantly outperforms all the baselines.

There are many possible extensions to this model, e.g. using the same techniques to jointly optimise choosing the number of attributes, aggregation, word choice, referring expressions, and so on, in a hierarchical manner.

We are currently collecting data in targeted Wizard-of-Oz experiments, to derive a fully data-driven training environment and test the learnt policy with real users, following (Rieser and Lemon, 2008b). The trained NLG strategy will also be integrated in an end-to-end statistical system within the CLASSiC project (www.classic-project.org).

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216594 (CLASSiC project project: www.classic-project.org) and from the EPSRC project no. EP/E019501/1.

References

A. Baddeley. 2001. Working memory and language: an overview. *Journal of Communication Disorder*, 36(3):189–208.

Vera Demberg and Johanna D. Moore. 2006. Information presentation in spoken dialogue systems. In *Proceedings of EACL*.

James Henderson, Oliver Lemon, and Kallirroi Georgila. 2008. Hybrid reinforcement / supervised learning of dialogue policies from fixed datasets. *Computational Linguistics (to appear)*.

Srinivasan Janarthanam and Oliver Lemon. 2008. User simulations for online adaptation and knowledge-alignment in Troubleshooting dialogue systems. In *Proc. of SEMdial*.

Alexander Koller and Ronald Petrick. 2008. Experiences with planning for natural language generation. In *ICAPS*.

Alexander Koller and Matthew Stone. 2007. Sentence generation as planning. In *Proceedings of ACL*.

Oliver Lemon. 2008. Adaptive Natural Language Generation in Dialogue using Reinforcement Learning. In *Proceedings of SEMdial*.

Johanna Moore, Mary Ellen Foster, Oliver Lemon, and Michael White. 2004. Generating tailored, comparative descriptions in spoken dialogue. In *Proc. FLAIRS*.

Crystal Nakatsu and Michael White. 2006. Learning to say it well: Reranking realizations by predicted synthesis quality. In *Proceedings of ACL*.

Alice Oh and Alexander Rudnicky. 2002. Stochastic natural language generation for spoken dialog systems. *Computer, Speech & Language*, 16(3/4):387–407.

Tim Paek and Eric Horvitz. 2000. Conversation as action under uncertainty. In *Proc. of the 16th Conference on Uncertainty in Artificial Intelligence*.

Joseph Polifroni and Marilyn Walker. 2008. Intensional Summaries as Cooperative Responses in Dialogue Automation and Evaluation. In *Proceedings of ACL*.

Verena Rieser and Oliver Lemon. 2008a. Does this list contain what you were searching for? Learning adaptive dialogue strategies for Interactive Question Answering. *J. Natural Language Engineering*, 15(1):55–72.

Verena Rieser and Oliver Lemon. 2008b. Learning Effective Multimodal Dialogue Strategies from Wizard-of-Oz data: Bootstrapping and Evaluation. In *Proceedings of ACL*.

S. Singh, D. Litman, M. Kearns, and M. Walker. 2002. Optimizing dialogue management with Reinforcement Learning: Experiments with the NJFun system. *JAIR*, 16:105–133.

Amanda Stent, Marilyn Walker, Steve Whittaker, and Preetam Maloor. 2002. User-tailored generation for spoken dialogue: an experiment. In *In Proc. of ICSLP*.

Amanda Stent, Rashmi Prasad, and Marilyn Walker. 2004. Trainable sentence planning for complex information presentation in spoken dialog systems. In *Association for Computational Linguistics*.

R. Sutton and A. Barto. 1998. *Reinforcement Learning*. MIT Press.

Marilyn A. Walker, Candace A. Kamm, and Diane J. Litman. 2000. Towards developing general models of usability with PARADISE. *Natural Language Engineering*, 6(3).

Marilyn Walker, S. Whittaker, A. Stent, P. Maloor, J. Moore, M. Johnston, and G. Vasireddy. 2004. User tailored generation in the match multimodal dialogue system. *Cognitive Science*, 28:811–840.

Marilyn Walker, Amanda Stent, François Mairesse, and Rashmi Prasad. 2007. Individual and domain adaptation in sentence planning for dialogue. *Journal of Artificial Intelligence Research (JAIR)*, 30:413–456.

Steve Whittaker, Marilyn Walker, and Johanna Moore. 2002. Fish or Fowl: A Wizard of Oz evaluation of dialogue strategies in the restaurant domain. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)*.

Stephen Whittaker, Marilyn Walker, and Preetam Maloor. 2003. Should i tell all? an experiment on conciseness in spoken dialogue. In *Proc. European Conference on Speech Processing (EUROSPEECH)*.

Andi Winterboer, Jiang Hu, Johanna D. Moore, and Clifford Nass. 2007. The influence of user tailoring and cognitive load on user performance in spoken dialogue systems. In *Proc. of the 10th International Conference of Spoken Language Processing (Interspeech/ICSLP)*.

SJ Young, J Schatzmann, K Weilhammer, and H Ye. 2007. The Hidden Information State Approach to Dialog Management. In *ICASSP 2007*.