Natural Language Generation as Incremental Planning Under Uncertainty: Adaptive Information Presentation for Statistical Dialogue Systems

Verena Rieser*, Oliver Lemon, and Simon Keizer.

***Abstract*—We present and evaluate a novel approach to natural language generation (NLG) in statistical spoken dialogue systems (SDS) using a data-driven statistical optimisation framework for incremental information presentation (IP), where there is a trade-off to be solved between presenting "enough" information to the user while keeping the utterances short and understandable. The trained IP model is adaptive to variation from the current generation context (e.g. a user and a non-deterministic sentence planner), and it incrementally adapts the IP policy at the turn level. Reinforcement learning is used to automatically optimise the IP policy with respect to a data-driven objective function. In a case study on presenting restaurant information, we show that an optimised IP strategy trained on WoZ data outperforms a baseline mimicking the wizard behaviour in terms of total reward gained. The policy is then also tested with real users, and improves on a conventional hand-coded IP strategy used in a deployed SDS in terms of overall task success. The evaluation found that the trained information presentation strategy significantly improves dialogue task completion for real users, with up to a 8.2% increase in task success. This methodology also provides new insights into the nature of the IP problem, which has previously been treated as a module following dialogue management with no access to lower-level context features (e.g. from a surface realiser and/or speech synthesiser).**

***Index Terms*—Information presentation, natural language generation, natural language user interfaces, reinforcement learning.**

I. 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 search results to the user can be realised as a *summary* [1], [2], or by *comparing* items [3], or by picking one item and *recommending* it to the user [4]. 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. [3], [5], user knowledge, e.g. [6], [7], or predicted TTS quality, e.g. [8], [9].

With spoken dialogue systems now employing more statistical, e.g. [10], and incremental techniques, e.g. [11], language generation faces new challenges. First, in fully statistical dialogue systems, all components can introduce uncertainty, i.e. other components cannot know for sure how "higher up" or "lower down" components in the dialogue system pipeline will perform, but may only have an estimate of their *likely* behaviour. See for example the system developed by the CLASSIC project[1] [12]. As such, the uncertainty the system has to deal with is not only restricted to automatic speech recognition (ASR), but all modules can introduce uncertainty. Strategy optimisation therefore needs to consider adaptation to uncertainty from input modules and the variation that is introduced by "lower level" modules such as utterance realisation and speech synthesis. For example, the NLG system might consider recognition confidence in slot values [13] as well as variations in predicted text-to-speech (TTS) quality [8], [9].

Secondly, NLG for incremental dialogue systems needs to be able to have the ability to accommodate user barge-ins [11]. Note that the timing of when to start generation is usually determined by the dialogue manager. However, barge-ins are controlled by the user and thus, incremental systems need to consider possible user barge-ins any time. If the user decides not to interrupt the system, NLG content selection has the choice whether to continue generation or whether to stop. As such, incremental planning for generation requires an estimate of "*what would happen if we stop generating now?*", i.e. an estimate of dynamically changing next user actions.

To deal with these requirements, we propose a new model which treats NLG as a statistical incremental planning problem, enabling automatic adaptation to the dialogue context, analogously to current statistical approaches to dialogue management (DM), e.g. [14], [15], [16], [17].

In NLG we have similar trade-offs and unpredictability as in dialogue management, and in some systems the content planning and DM tasks are combined. For example, on the one hand, very long system utterances should be avoided, because users may become confused or impatient, but on the other hand, each individual part of the utterance will convey some (potentially) useful information to the user. There is therefore an optimisation problem to be solved. Moreover, this optimisation has to deal with uncertainty, as the user judgements or the (most likely) user reaction to each NLG action are unpredictable, and the behaviour of subsequent modules, such as the surface realiser, may also be variable.

In this paper we present and evaluate a novel framework for adaptive natural language generation where the problem is formulated as incremental decision making under uncertainty,

School of Mathematical and Computer Sciences (MACS), Heriot-Watt University, Edinburgh, EH14 4AS UK e-mail: (see http://www.macs.hw.ac.uk/InteractionLab).

Manuscript received January 18, 2013; revised July 1st, 2013.

[1] http://www.classic-project.org/

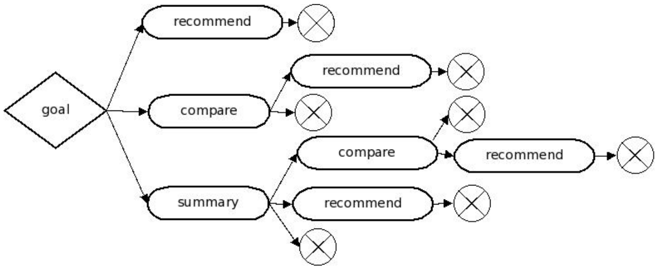


Fig. 1. Possible NLG policies (X=stop generation)

which can be approached using statistical planning methods [18], [19], [20], [21]. This model is also being explored by other researchers [22], [23], [24], [25]. We have applied the theory to a variety of NLG problems, such as referring expression generation [6], [7] and incremental dialogue phenomena, such as barge-in [13], [26]. Here, we focus on adaptive information presentation (IP) in statistical spoken dialogue systems.

The remainder of the paper proceeds as follows. In Section II we introduce the information presentation problem. In Section III we present the general framework of NLG as planning under uncertainty. In Section IV we describe a Wizard-of-Oz (WoZ) data collection. In Section V we explain how we build a simulated training environment from this data. In Section VI we present results from training and testing in simulation and in Section VII we present results from a user study. Section VIII concludes with a discussion of future directions.

II. Previous work: information presentation in SDS

Work on evaluating SDS suggests that the information presentation (IP) phase is the primary contributor to dialogue duration [27], and as such, is a central aspect of SDS design. During this phase the system returns a set of items ("hits") from a database, which match the user's current search constraints. An inherent problem in this task is the trade-off between presenting "enough" information to the user (for example helping them to feel confident that they have a good overview of the search results) versus keeping the utterances short and understandable, see for example [28], [29].

Broadly speaking, IP for SDS can be divided into two main steps: 1) IP strategy selection and 2) Content or attribute selection, which again can be divided into attribute ranking (according to a user model) and the number of attributes to be selected. In this work we will concentrate on strategy selection as well as selecting the number of attributes from a list of ranked attributes, which we assume to be already provided by a given user model.

Prior work has presented a variety of *IP strategies* for structuring information (see examples in Table I). For example, the SUMMARY strategy is used to guide the user's attention to a relevant subset of search results, i.e. it draws the user's attention to relevant attributes by grouping the current results from the database into clusters, e.g. [1], [2]. Other studies investigate a COMPARE strategy, e.g. [30], [31], while much work in SDS uses a RECOMMEND strategy, e.g. [4], [32].

In a previous proof-of-concept study [19] we showed that each of these strategies has its own strengths and drawbacks, dependent on the particular context in which information needs to be presented to a user. Here, we will also explore possible (incremental) combinations of the strategies, for example SUMMARY followed by RECOMMEND, as also explored by [33], see Figure 1.

Prior work on *content or attribute Selection* has used a "Summarize and Refine" approach [1], [34] to determine which attributes should be used. This method employs utility-based attribute selection with respect to how each attribute (e.g. price or food type in restaurant search) of a set of items helps to narrow down the user's goal to a single item. Related work explores a user modelling approach, where attributes are ranked according to user preferences [5], [35], [36]. Our data collection (see Section IV) and training environment (Section V) incorporate these approaches.

The work in this paper is the first to apply a data-driven method to this whole decision space, i.e. combinations of information presentation strategies as well as selecting the number of attributes. We also show the utility of both lower-level features (e.g. from the surface realiser) and higher-level features (e.g. from dialogue management) for this problem. Previous work has focused on individual aspects of the problem (e.g. how many attributes to generate, or when to use a SUMMARY), using a pipeline model for SDS with DM features as input, and where NLG has no knowledge of lower-level features, e.g. lower-level surface realization and associated features such as sentence length.

III. Natural Language Generation as Planning Under Uncertainty

We adopt the general framework of NLG as planning under uncertainty (see [18], [19] for the initial version of this approach). Some aspects of NLG have been treated as planning, e.g. [37], but not as statistical planning.

Within an SDS architecture, NLG actions take place in a stochastic environment, consisting for example of a user and a stochastic realiser, where the individual NLG actions have uncertain effects on the environment. For example, presenting differing numbers of attributes to the user, makes the user more or less likely to choose an item, as shown by [38] for multi-modal interaction.

Most SDSs employ fixed template-based generation. Our model, however, employs a non-deterministic sentence planner and surface realiser for SDS, based on [39]. This introduces additional variation, to which higher level NLG decisions will need to react. 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 summarise all the items and then to recommend the highest ranking one. Each such action has uncertain effects due to the stochastic realiser. For example, the realiser might generate a different sentence structure or employ different numbers of attributes depending on its own processing constraints (see e.g. the realiser used to collect the

TABLE I
EXAMPLE REALISATIONS, GENERATED WHEN THE USER PROVIDED cuisine=Indian, AND WHERE THE WIZARD HAS ALSO SELECTED THE ADDITIONAL ATTRIBUTE price FOR PRESENTATION TO THE USER.

Strategy	Example utterance
SUMMARY no user model	I found 26 restaurants, which have Indian cuisine. 11 of the restaurants are in the expensive price range. Furthermore, 10 of the restaurants are in the cheap price range and 5 of the restaurants are in the moderate price range.
SUMMARY with user model	26 restaurants meet your query. There are 10 restaurants which serve Indian food and are in the cheap price range. There are also 16 others which are more expensive.
COMPARE by item	The restaurant called Kebab Mahal is an Indian restaurant. It is in the cheap price range. And the restaurant called Khushi's, which is also an Indian restaurant, is in the moderate price range.
COMPARE by attribute	The restaurant called Kebab Mahal and the restaurant called Khushi's are both Indian restaurants. However, Kebab Mahal is in the cheap price range while Khushi's is moderately priced.
RECOMMEND	The restaurant called Kebab Mahal has the best overall quality amongst the matching restaurants. It is an Indian restaurant, and it is in the cheap price range.

MATCH project data, see [40] and [41]). 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 the context of an interactive, stochastic, system, thus has the following important characteristics 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*.

As such, the problem of planning how to (incrementally) generate an utterance for SDS falls naturally into the class of statistical planning problems, rather than rule-based approaches such as [42], [3], or supervised learning as explored in previous work, such as classifier learning and re-ranking, e.g. [39], [43], [44]. These supervised approaches involve the ranking of a set of completed plans/utterances and do not adapt online to the context or the user. reinforcement learning (RL) provides a principled, data-driven optimisation framework for our type of planning problem maximising some notion of long-term reward or utility [45]. Note that there also is closely related work which also explores NLG as a process of maximising utility using Game Theory [46], [47].

In the following we further illustrate this approach using a worked example of information presentation.

A. Worked Example

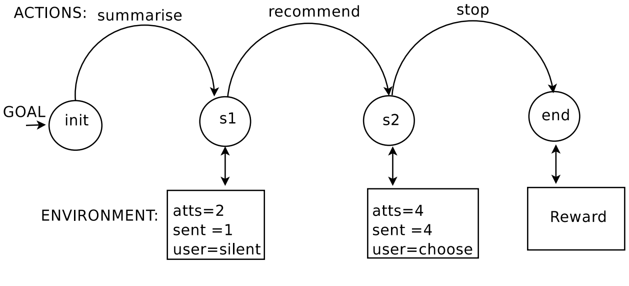


Fig. 2. Example RL-NLG action sequence for Table II.

The input to the NLG planning 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, in the case of fully stochastic systems, variation from a stochastic sentence realiser, e.g. systems such as SPaRKy [39].

Now, let us walk through one simple utterance plan for IP strategy selection only, as carried out by this model, as shown in Table II and Figure 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*). In this example, the policy chooses the action SUMMARY and this transitions us to state $s1$, where we observe that in this example 2 attributes and 1 sentence have been generated by lower level modules. Also, the user is predicted to remain silent. In this state, the current NLG policy chooses to next RECOMMEND the top ranked item (i_5, for this user), which takes us to state $s2$, where, by now, a total of 4 attributes have been generated in a total of 4 sentences, and the user is now predicted to choose an item. As such, the policy determines that in states like $s2$ the 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 V-C), and used to update the NLG policy in each of the visited state-action pairs via back-propagation.

In Section VI we present a model which integrates strategy and attribute selection into a hierarchical learning architecture, i.e. the number of attributes selected is also optimised with respect to the expected reward.

IV. WIZARD-OF-OZ DATA COLLECTION

In a previous proof-of-concept study [19] using two data sets made available by the MATCH project, see [40] and [41] we showed that each of the IP strategies in Table I has its own

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASL.2014.2315271, IEEE Transactions on Audio, Speech, and Language Processing

TABLE II
EXAMPLE UTTERANCE PLANNING SEQUENCE FOR FIGURE 2.

State	Action	Example output	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)	*I found 4 moderately priced Thai restaurants.*	att=2, sent=1, user=silent
s2	RL-NLG: RECOMMEND(i_5)	*I can recommend Lemongrass. It is highly rated for user service. It is located in Tollcross.*	att=4, sent=4, user=choose(i_5)
end	RL-NLG: stop		calculate Reward

strengths and drawbacks, dependent on the particular context in which information needs to be presented to a user. In the following we explore possible combinations of the strategies in a WoZ data collection. In this WoZ study, we asked humans (our “wizards”) to produce appropriate IP actions in different dialogue contexts, when interacting with other humans (the “users”), who were told that they were talking to an automated SDS. The wizards were experienced researchers in SDS and were familiar with the search domain (i.e. restaurants in Edinburgh). They were instructed to select IP structures and attributes for NLG so as to most efficiently allow users to find a restaurant matching their search constraints, where we used a database of 340 Edinburgh restaurants[2] provided by TheList[3].

The task for the wizards was to decide which IP structure to use next (see Section IV-B for a list of IP strategies to choose from), how many attributes to mention (e.g. cuisine, price range, location, food quality, and/or service quality), and whether to stop generating. Our hypothesis is that these choices depend on varying numbers of database matches, varying prompt realisations, and varying user behaviour. Wizard utterances were synthesised using the Cereproc text-to-speech engine[4] . The user speech input was delivered to the wizard using Voice Over IP.

In order to make the wizards’ decisions comparable with real system decisions, the wizard only sees what the systems would have as input, i.e. the experimenter will listen to the user’s utterance and transcribe it into a simple semantic representation of slot-value pairs. We also experimented with introducing noise to simulate speech recognition errors, following a similar method to [49]. However, noise did not seem to have a significant effect on either wizard or user behaviour [50].

Figure 3 shows the web-based interface for the wizard. The wizard GUI contains 5 main panels:

A: The wizard receives the user’s query as attribute values. The experimenter has a similar input panel. There are 5 searchable attributes in total, which can also be negative (“not expensive”).

B: The retrieved database items are presented in an ordered list. We use a user modelling approach for ranking the restaurants, where attributes are ranked according to user preferences. We assume a default user who cares about cheap food with high quality and good service.

C: The wizard then chooses which strategy and which attributes to generate next, by clicking radio buttons. The attribute/s specified in the last user query are pre-selected by default. The strategies can only be combined in the orders as specified in Figure 1.

D: An utterance is automatically generated by the NLG stochastic surface realiser (see Section IV-B) every time the wizard selects a strategy, and is displayed in an intermediate text panel.

E: The wizard can decide to add the generated utterance to the final output panel. The text in the final panel is sent to the user via TTS, once the wizard decides to stop generating.

A. Experimental Setup and Data collection

We collected 213 dialogues with 18 subjects and 2 wizards [51]. Each user performed a total of 12 tasks, where no task set was seen twice by any one wizard. Note that 3 dialogues were discharged due to technical difficulties. The majority of users were from a range of backgrounds in a higher education institute, in the age range 20-30, native speakers of English, and none had any prior experience of spoken dialogue systems. After each task, the user answered a questionnaire on a 6 point Likert scale, regarding the perceived generation quality in that task. The wizards’ IP strategies were highly ranked by the users on average (4.7), and users were able to select a restaurant in 98.6% of the cases. Also, no significant difference between the wizards was observed.

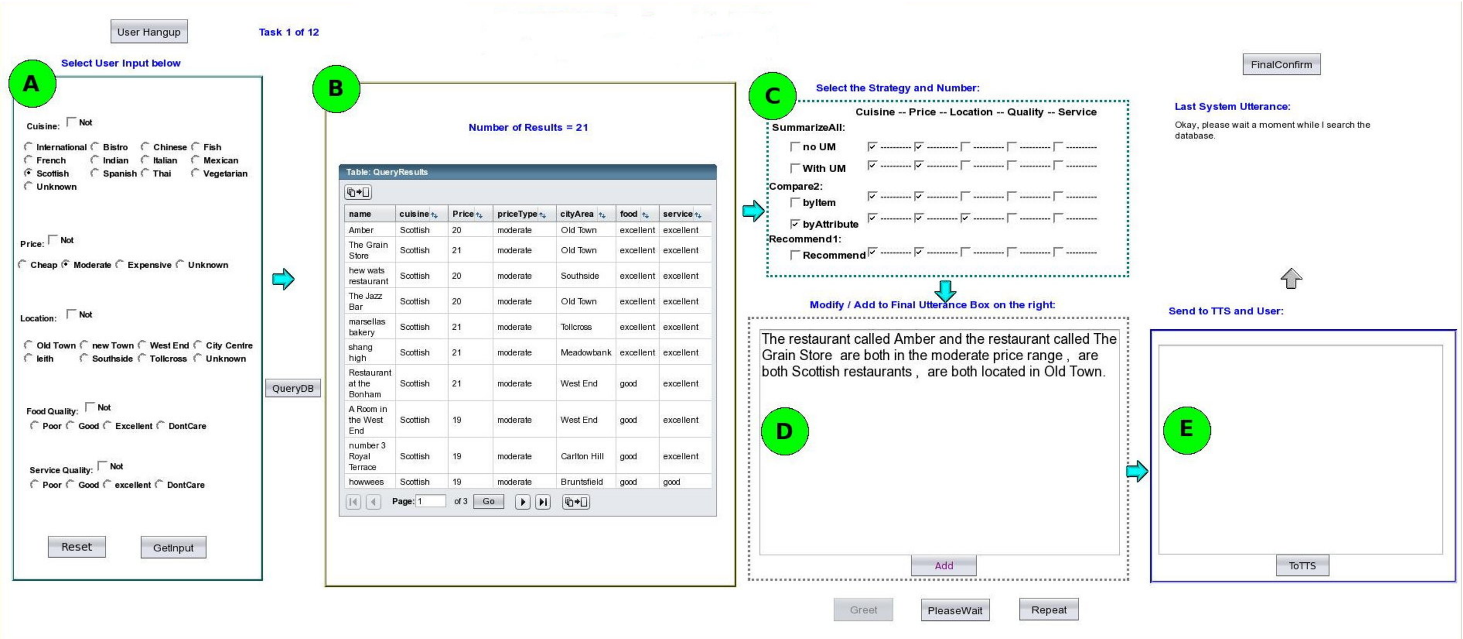
The data contains 2236 utterances in total: 1465 wizard utterances and 771 user utterances. We automatically extracted 81 features (e.g #sentences, #DBhits, #turns, #ellipsis)[5] from the XML logfiles after each dialogue. These features can be categorised into 7 broader categories: (1) general information (containing 9 features), e.g. user information; (2) turn

[2] Note that our approach is scalable to larger database sizes. The number of retrieved database items currently serves as a feature in the state space as well as a reward feature, where presenting large sub-sets of database items is negatively rewarded (see Section V-C). By increasing the number of items in the database, the size of the state space therefore increases, and learning would take more iterations. This feature could be quantised, or automatic state-space compression techniques could be used to maintain tractability [48].

[3] http://www.list.co.uk/

[4] http://www.cereproc.com/

[5] The full corpus and a list of features is available from the CLASSiC project data repository, see http://www.macs.hw.ac.uk/iLabArchive/CLASSiCProject/Data/myaccount.php



Cuisine: □ Not
○ International ○ Bistro ○ Chinese ○ Fish
○ French ○ Indian ○ Italian ○ Mexican
● Scottish ○ Spanish ○ Thai ○ Vegetarian
○ Unknown

Price: □ Not
○ Cheap ● Moderate ○ Expensive ○ Unknown

Location: □ Not
○ Old Town ○ new Town ○ West End ○ City Centre
○ leith ○ Southside ○ Tollcross ○ Unknown

Food Quality: □ Not
○ Poor ○ Good ○ Excellent ○ DontCare

Service Quality: □ Not
○ Poor ○ Good ○ excellent ○ DontCare

Number of Results = 21

Table: QueryResults

name	cuisine	Price	priceType	cityArea	food	service
Amber	Scottish	20	moderate	Old Town	excellent	excellent
The Grain Store	Scottish	21	moderate	Old Town	excellent	excellent
hew wats restaurant	Scottish	20	moderate	Southside	excellent	excellent
The Jazz Bar	Scottish	20	moderate	Old Town	excellent	excellent
marsellas bakery	Scottish	21	moderate	Tollcross	excellent	excellent
shang high	Scottish	21	moderate	Meadowbank	excellent	excellent
Restaurant at the Bonham	Scottish	21	moderate	West End	good	excellent
A Room in the West End	Scottish	19	moderate	West End	good	excellent
number 3 Royal Terrace	Scottish	19	moderate	Carlton Hill	good	excellent
howwees	Scottish	19	moderate	Bruntsfield	good	good

Page: 1 of 3

Select the Strategy and Number:

	Cuisine	Price	Location	Quality	Service
SummarizeAll:					
□ no UM	☑	☑	□	□	□
□ With UM	☑	☑	□	□	□
Compare2:					
□ byItem	☑	☑	□	□	□
☑ byAttribute	☑	☑	☑	□	□
Recommend1:					
□ Recommend	☑	☑	□	□	□

Last System Utterance:

Okay, please wait a moment while I search the database.

Modify / Add to Final Utterance Box on the right:

The restaurant called Amber and the restaurant called The Grain Store are both in the moderate price range , are both Scottish restaurants , are both located in Old Town.

Send to TTS and User:

Fig. 3. Wizard interface.

information (4 features), e.g. turn number; (3) NLG related information (14 features), e.g. the chosen IP strategy; (4) task-based information (22 features), e.g. slot values; (5) features from the annotated user reply (4 features), e.g. user dialogue act; (6) features describing the artificially introduced noise (16 features) and (7) user questionnaire ratings per task & objective measures (12 features). The user questionnaire was similar to the one described in Section VII-B, with additional questions focussing on the NLG quality, such as *"The way the system presented information was good. "*, *"The system's utterances had the right length."*, *"The system gave me a good overview of all the available options."*. A full list of annotated features is described in [50].

B. NLG Realiser

We implemented a NLG realiser for the chosen IP structures and attribute choices, in order to realise the wizards' choices in real time. This generator is based on data from the stochastic sentence planner SPaRKy [39]. We replicated the variation observed in SPaRKy by analysing high-ranking example outputs (given the highest possible score by the SPaRKy judges) and implemented the variance using stochastic sentence generation templates:[6] The realisations vary in sentence aggregation, aggregation operators (e.g. 'and', period, or ellipsis), contrasts (e.g. 'however', 'on the other hand') and referring expressions (e.g. 'it', 'this restaurant') used. The length of an utterance also depends on the number of attributes chosen, i.e. the more attributes the longer the utterance. All of these variations were logged.

The templates pre-define the sentence structure with place-holders for the actual values. For example, for comparing two items by `cuisine`, `price` and `location`, the following templates could be used:

(T1) *The restaurant called* `<name1>` *is* `<a/n>` `<cuisine1>` *restaurant which is in the* `<price1>` *price range and it is located in* `<location1>` `<CONTRAST>` *the restaurant called* `<name2>` *is* `<a/n>` `<cuisine2>` *restaurant which is in the* `<price2>` *price range and it is located in* `<location2>`.

(T2) `<name1>` *is* `<a/n>` `<cuisine1>` *restaurant* `<CONTRAST>` `<name2>` *is* `<a/n>` `<cuisine2>` *restaurant.* `<name1>` *is in the* `<price1>` *price range* `<CONTRAST>` `<name2>` *is in the* `<price2>` *price range.* `<name1>` *is located in* `<location1>` `<CONTRAST>` `<name2>` *is located in* `<location2>`.

Where `<CONTRAST>` is randomly chosen from the following set of contrastive connectives: { `[while],[. ], [. However], [, and], [. On the other hand], [but ]` }, following [31]. In total, we pre-defined over 90 templates, with about 30 for each of the following three IP strategies, where one was chosen at random during generation.

- SUMMARY of all matching restaurants using a user model (UM) approach, following [1], where attributes are ranked and clustered according to (pre-defined) user preferences.
- COMPARE the top 2 restaurants by Item (i.e. listing all the attributes for the first item and then for the other) or by Attribute (i.e. directly comparing the different attribute values).
- RECOMMEND the top-ranking restaurant (according to a UM).

[6] Note that we decided against using the freely available Java implementation of SPaRKy (jSPaRKy v 2.0). First, jSPaRKy generates utterances in past tense (presumably because it is trained on the Penn Tree Bank). Second, the introduced disfluencies might have had a negative impact on the user ratings.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASL.2014.2315271, IEEE Transactions on Audio, Speech, and Language Processing

Note that there was no discernible pattern in the data about the wizards' decisions between the UM/no UM and the byItem/byAttribute versions of the strategies. In this study we therefore concentrate on the higher level decisions (SUMMARY VS. COMPARE VS. RECOMMEND) and model UM/no UM and the byItem/byAttribute versions as variations introduced by the realiser.

C. Supervised Baseline strategy

We analysed the WoZ data to explore the strategies that human wizards explored for this task.Note that while the wizards showed significantly different behaviour in terms of overall frequency, however no significant difference in user ratings was detected [50]. We therefore hypothesize that wizards select different strategies according to different contexts. 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. We also observed that there was a wide spread user ratings for different dialogues. We therefore took the top rated 50% of the IP instances ($n = 205$) to build a baseline model which reflects "good" wizard behaviour.

We used a variety of supervised learning methods to create a model of the highly rated wizard behaviour. Please see [50] for further details. The best performing method was Rule Induction (JRip). [7] The model achieved an accuracy of 43.19% which is significantly ($p < .001$) better than the majority baseline of always choosing SUMMARY (34.65%). [8] The resulting rule set is shown in Figure 4, where the determining features are number of database hits retrieved `dbHits` and the IP strategy realised in the previous turn `prevNLG`:

```
IF (dbHits = 1):
   THEN nlgStrategy = recommend;
IF (dbHits <= 9) & (prevNLG = summary):
   THEN nlgStrategy = compare;
IF(dbHits >= 10) & (prevNLG = summary):
   THEN  nlgStrategy = recommend;
ELSE nlgStrategy = summary;
```

Fig. 4. Rules learned by JRip for the wizard model ('`dbHits`'= number of database matches, '`prevNLG`'= previous NLG action)

The features selected by this model were only "high-level" features, i.e. the features that an IP module receives as input from a Dialogue Manager. We further analysed the importance of different features using feature ranking and selection methods [50], finding that the human wizards (in this specific setup) did not pay significant attention to any lower level features, e.g. variation from surface realisation.

Despite the simplicity of the learned supervised policy, it achieves up to 87.6% of the possible reward on this task, as we show in Section VI-B, and so can be considered a serious baseline against which to measure performance. Below, we will show that reinforcement learning produces a significant improvement over the strategies present in the original data, especially in cases where RL has access to "lower level" features of the context. This confirms previous findings that human wizard behaviour is not necessarily "optimal" [38].

We also learn a supervised model for selecting the number of attributes, trying a variety of SL techniques. However, the learned models did not show any significant improvements over a majority or a random baseline, with the majority baseline performing best. We therefore implemented a majority baseline (randomly choosing between 3 or 4 attributes) as a baseline system.

V. SIMULATED ENVIRONMENT FOR LEARNING

Here we "bootstrap" a simulated training environment from the WoZ data, following [17], [53].

A. User Simulations

User simulations are commonly used to train strategies for Dialogue Management, see for example [4]. A user simulation for (incremental) NLG is very similar, in that it is a predictive model of the most likely next user act. [9] However, this NLG predicted 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*.

We are mainly interested in the following user reactions:

1) `select`: the user chooses one of the presented items, e.g. *"Yes, I'll take that one."* This reply type indicates that the information presentation was sufficient for the user to make a choice.
2) `addInfo`: The user provides more attributes, e.g. *"I want something cheap."* This reply type indicates that the user has more specific requests, which s/he wants to specify after being presented with the current information.
3) `requestMoreInfo`: The user asks for more information, e.g. *"Can you recommend me one?", "What is the price range of the last item?"* This reply type indicates that the system failed to present the information the user was looking for.
4) `askRepeat`: The user asks the system to repeat the same message again, e.g. *"Can you repeat?"* This reply type indicates that the utterance was either too long or confusing for the user to remember, or the TTS quality was not good enough, or both.
5) `silence`: The user does not say anything. In this case it is up to the system to take initiative.
6) `hangup`: The user closes the interaction.

We built user simulations using n-gram models of system (s) and user (u) acts, as first introduced by [56]. In order

[7] The WEKA implementation of [52]'s RIPPER.

[8] We believe that the relatively low model accuracy is due to data sparsity and diverse behaviour of the wizards. We hypothesize that multiple different generation actions are viable at each point and so while the model does not predict the exact pattern in the data it predicts a reasonable pattern. This hypothesis is supported later on by the high reward it gains. Note that the reward measures how "appropriate" an action is in a specific context (see Section V-C).

[9] Similar to the internal user models applied in recent work on POMDP (Partially Observable Markov decision process) dialogue managers [4], [54], [55] for estimation of user act probabilities.

TABLE III
KULLBACK-LEIBLER DIVERGENCE FOR THE DIFFERENT USER SIMULATIONS (US), HIGHLIGHTING *worst scoring* AND **best scoring** METHODS.

discounting method	bi-gram US	extended bi-gram US
Witten-Bell	0.086	*0.512*
Good-Turing	0.086	**0.163**
absolute	*0.091*	0.246
linear	**0.011**	0.276

to account for data sparsity, we apply different *discounting* ("smoothing") techniques including *back-off*, using the CMU Statistical Language Modelling toolkit [57], see Table III. We construct a **bi-gram** model for the users' reactions to the system's IP structure decisions ($P(a_{u,t}|IP_{s,t})$), and an **extended bi-gram** (i.e. IP structure + attribute choice) model for predicting user reactions to the system's combined IP structure and attribute selection decisions: $P(a_{u,t}|IP_{s,t}, attributes_{s,t})$, where $a_{u,t}$ is the predicted next user action at time t, $IP_{s,t}$ was the system's information presentation action at t, and $attributes_{s,t}$ is the attributes selected by the system at t.

We evaluated the performance of these models by measuring dialogue similarity to the original data, based on the Kullback-Leibler (KL) divergence, as also used by, e.g. [58], [59], [60]. We compare the raw probabilities as observed in the data with the probabilities generated by our n-gram models using different discounting techniques, including Witten-Bell, Good-Turing, absolute and linear discounting [57], see Table III. All the models have a small divergence from the original data (especially the bi-gram model), suggesting that they are reasonable simulations for training and testing NLG policies.

The absolute discounting method for the bi-gram model is most dissimilar to the data, as is the Witten-Bell method for the extended bi-gram model, i.e. the models using these discounting methods have the highest KL score. The best performing methods (i.e. most similar to the original data), are linear discounting for the bi-gram model and Good-Turing for the extended bi-gram. We use the most similar user simulations for system training, and the most dissimilar user simulations for testing NLG policies, in order to test whether the learned policies are robust and adaptive to unseen dialogue contexts.

B. Database matches and Attention Modelling

An important task of information presentation is to support the user in choosing between all the available items (and ultimately in selecting the most suitable one) by structuring the current information returned from the database, as explained in Section II. We therefore represent the items brought to the user's attention as a feature in our learning experiments. In particular, attention modelling is used to determine the number of database hits ($\#DBhits$) used in the reward function, see Section V-C. This feature reflects how the different IP strategies structure information with different numbers of attributes, similar to the clustering approach discussed in [5]. We implement this shift of the user's attention focus analogously to discovering the user's goal in dialogue management: every time the predicted next user act is to add information (addInfo), we infer that the user is therefore only interested in a subset of the previously presented results and so the system will focus on this new subset of database items in the rest of the generated utterance. For example in Table I, the user's attention reduces from a total of 26 $DBhits$ to 10 after the SUMMARY (with UM) since the user is only interested in cheap, Indian places.

C. Data-driven Reward function

The reward/evaluation function $\mathcal{R}$ is constructed from the WoZ data, using a stepwise linear regression, following the PARADISE framework [61]. This model selects the context features (see Section IV-A) which significantly influenced the users' ratings for the NLG strategy in the WoZ questionnaire. We also assign a value to the user's reactions ($valueUserReaction$), similar to optimising task success for DM [4]. In [50] we showed that different user replies are positively or negatively correlated with user ratings. The numerical values were manually assigned to reflect this. In principle it would also be possible to learn the user reaction value function. The agent receives

$$valueUserReaction = \begin{cases} +100 & \text{if the user selects an item,} \\ 0 & \text{if the user adds further constraints to the search,} \\ \text{-100} & \text{if the user does something else} \end{cases} \quad (1)$$

The agent is encouraged to choose those sequences of actions that lead to the user selecting a restaurant as quickly as possible. We then run a stepwise linear regression on the annotated WoZ data (relating contextual features to user task ratings), resulting in the model in Equation 2 ($R^2 = .26$). The chosen features indicate that users' ratings are influenced by higher level and lower dialogue level features: Users like to be focused on a small set of database hits (where $\#DBhits$ ranges over [1-100]), which will enable them to choose an item ($valueUserReaction$), while keeping the IP utterances short (where $\#sentence$ is in the range [2-18]):

$$\begin{aligned} \mathcal{R} = -1.2 \times \#DBhits \\ +0.121 \times valueUserReaction \\ -1.43 \times \#sentence \end{aligned} \quad (2)$$

The reward is calculated at the end of an episode, which for this task constitutes a full NLG utterance, also see Section III-A. Note that the worst possible reward for a NLG move is therefore $(-1.20 \times 100) - (.121 \times 100) - (18 \times 1.43) = -157.84$. This is achieved by presenting 100 items to the user in 18 sentences[10], in such a way that the user immediately ends the conversation unsuccessfully. The top possible reward is achieved in the rare cases where the system can immediately present one item to the user using a minimum of two sentences, and the user then selects that item, i.e. $\mathcal{R}$ = $-(1.20 \times 1) + (.121 \times 100) - (2 \times 1.43) = 8.06$.

[10] Note that the maximum possible number of sentences generated by the realiser is 18 for the full IP sequence SUMMARY+COMPARE+RECOMMEND using all the attributes.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASL.2014.2315271, IEEE Transactions on Audio, Speech, and Language Processing

VI. Training and Testing the Learned Policies in Simulation

We now formulate the problem as a Markov decision process (MDP), where states are NLG dialogue contexts and actions are NLG decisions. 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. the user simulation and realiser), and explicitly captures the uncertainty in the generation 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} as defined above, describing how good the result of action a was when performed in state s. The aim of the MDP is to maximise the long-term expected reward of its decisions, resulting in a *policy* which maps each possible state to an appropriate action in that state.

We then treat IP as a *hierarchical* joint optimisation problem, where first one of the possible single IP strategies (1-3) is chosen and then the number of attributes is decided, as shown in Figure 5. At each generation step, the MDP can choose 1-5 attributes (e.g. cuisine, price range, location, food quality, and/or service quality). Generation stops as soon as the user is predicted to select an item, i.e. the IP task is successful. (Note that the same constraint is operational for the WoZ baseline.)

$$\left[\begin{array}{ll}
\text{ACTIONS:} & \left[\begin{array}{l}
\textit{IP:}\left\{\begin{array}{l}\text{SUMMARY}\\ \text{COMPARE}\\ \text{RECOMMEND}\end{array}\right\}\\
\textit{\#attr:}\left\{\texttt{1,2,3,4,5}\right\}
\end{array}\right]\\
\text{STATE:} & \left[\begin{array}{l}
\texttt{statusCuisine}\left\{\text{0=unfilled,1=filled}\right\}\\
\texttt{statusQuality}\left\{\text{0=unfilled,1=filled}\right\}\\
\texttt{statusLocation}\left\{\text{0=unfilled,1=filled}\right\}\\
\texttt{statusPrice}\left\{\text{0=unfilled,1=filled}\right\}\\
\texttt{statusService}\left\{\text{0=unfilled,1=filled}\right\}\\
\texttt{\#sentence:}\left\{\texttt{2-18}\right\}\\
\texttt{\#dbHits:}\left\{\texttt{1-100}\right\}\\
\texttt{userSelect:}\left\{\text{0=false,1=true}\right\}\\
\texttt{userAddInfo:}\left\{\text{0=false,1=true}\right\}\\
\texttt{userElse:}\left\{\text{0=false,1=true}\right\}
\end{array}\right]
\end{array}\right]$$

Fig. 5. State-action space for the RL-NLG problem

States are represented as sets of NLG dialogue context features, see Figure 5. The state space comprises "lower-level" features about the realiser behaviour, including two features representing the number of attributes ($\#attr$) and sentences generated so far ($\#sentence$), and three binary features representing the user's predicted next action, as well as "high-level" features provided by the DM, including the current database hits brought to the user's attention ($\#dbHits$), the slots filled so far ($statusSlot$) and a subset of user actions (also see Section V-A). According to the reward function (Section V-C), we mainly care about whether the user selects (`select`) or adds information (`addInfo`). All other user acts (e.g. `requestMoreInfo`, `askRepeat`, or `silence`) are mapped to one state feature ($userElse$).

We trained the policy using a hierarchical SARSA algorithm [62] with linear function approximation [45], and the simulation environment described in Section V. The policy was trained for 60,000 iterations.

A. Experimental Set-up

We compare the learned strategies against the *supervised wizard baseline* as described in Section IV-C. For training, we use the user simulation model most similar to the data. For testing, we use a different user simulation model (the one which is most dissimilar to the data), see Section V-A.

We first investigate how well IP structure (without attribute choice) can be learned in increasingly complex **generation scenarios**, with respect to action choices and uncertainty in the environment. A generation scenario is a combination of a particular kind of NLG surface realiser (template vs. stochastic) along with different levels of variation introduced by certain features of the dialogue system. In general, the stochastic realiser introduces more variation in sentence length than the template-based realiser. We therefore investigate the following cases, starting with optimising IP structure choice only:

1.1 **IP structure choice wrt. template realiser:** Predicted next user action varies according to the bi-gram model ($P(a_{u,t}|IP_{s,t})$); Number of sentences and attributes per IP strategy is set by defaults, reflecting a template-based realiser.

1.2 **IP structure choice wrt. stochastic realiser:** IP structure where number of attributes per NLG turn is given at the beginning of each episode (e.g. set by the DM); Sentence generation according to the SPaRKy stochastic realiser model as described in Section IV-B.

We then investigate different scenarios for *jointly* optimising IP structure and attribute selection (Attr) decisions.

2.1 **IP structure choice + #attribute choice wrt. template realiser:** Predicted next user action varies according to extended bi-gram ($P(a_{u,t}|IP_{s,t}, attributes_{s,t})$) model; Number of sentences per IP structure set to default.

2.2 **IP structure choice + #attribute choice wrt. template realiser + Attention:** Extended bi-gram user simulation with Template realiser and attention model with respect to #DBhits and #attributes as described in Section V-B.

2.3 **IP structure choice + #attribute choice wrt. stochastic realiser** Extended bo-gram user simulation with sentence/attribute relationship according to stochastic realiser as described in Section IV-B.

2.4 **IP structure choice + #attribute choice wrt. stochastic realiser + Attention:** i.e. the full model = Predicted next user action varies according to extended bi-gram model+ attention model + Sentence/attribute relationship according to stochastic realiser.

TABLE V
IP STRATEGIES LEARNED FOR THE DIFFERENT SCENARIOS, WHERE (n) DENOTES THE NUMBER OF ATTRIBUTES GENERATED.

Scenario	IP strategies learned
1.1	All possible combinations generated.
1.2	All possible combinations generated.

2.1	RECOMMEND(5)
	SUMMARY(2)
	SUMMARY(2)+COMPARE(4)
	SUMMARY(2)+COMPARE(1)
	SUMMARY(2)+COMPARE(4)+RECOMMEND(5)
	SUMMARY(2)+COMPARE(1)+RECOMMEND(5)

2.2	RECOMMEND(5)
	SUMMARY(4)
	SUMMARY(4)+RECOMMEND(5)

2.3	RECOMMEND(2)
	SUMMARY(1)
	SUMMARY(1)+COMPARE(4)
	SUMMARY(1)+COMPARE(1)
	SUMMARY(1)+COMPARE(4)+RECOMMEND(2)

2.4	RECOMMEND(2)
	SUMMARY(2)
	SUMMARY(2)+COMPARE(4)
	SUMMARY(2)+RECOMMEND(2)
	SUMMARY(2)+COMPARE(4)+RECOMMEND(2)
	SUMMARY(2)+COMPARE(1)+RECOMMEND(2)

B. Results

We compare the average final reward (see Equation 2) gained by the baseline against the trained RL policies in the different scenarios for each 1000 test runs, using a paired samples t-test. The results are shown in Table IV. In 5 out of 6 scenarios the RL policy significantly ($p < .001$) outperforms the supervised baseline. We also report on the percentage of the top possible reward , and the raw percentage improvement of the RL policy. Note that the best possible reward ($\mathcal{R} = 8.06 = 100\%$) can only be gained in rare cases (see Section V-C).

The more complex the scenario, the harder it is to gain higher rewards for the policies in general (as more variation is introduced). However, the relative improvement in rewards also increases with complexity: RL learns to adapt to the more challenging scenarios. Note that the baseline does reasonably well in scenarios with variation introduced by only higher level features (e.g. scenario 2.2), however it fails to adapt well to lower level features (see Section IV-C). From these results we can infer that adapting to lower level features is important in gaining significant improvements over the baseline.

An overview of the range of different IP strategies learned for each scenario can be found in Table V, which is generated by analysing the respective test runs.

C. Learned Behaviour

Note that these strategies are context-dependent: the learner chooses how to proceed dependent on the features in the state space at each generation step. In the following we provide illustrative examples of what was learned for each of the generation scenarios, which we extracted from the respective test runs.

a) Scenario 1.1-1.2: IP strategy selection: The RL policy for Scenario 1.1 learned to start with a SUMMARY if the initial number of items returned from the database is high (>30). It will then stop generating if the user is predicted to select an item. Otherwise, it continues with a RECOMMEND. If the number of database items is low, it will start with a COMPARE and then continue with a RECOMMEND, unless the user selects an item.

In addition, the RL policy for Scenario 1.2 learns to adapt to a more complex scenario: the number of attributes requested by the DM and produced by the stochastic sentence realiser. It learns to generate the whole sequence (SUMMARY+COMPARE+RECOMMEND) if *#attributes* is low (<3), when the overall generated utterance (final *#sentences*) is still relatively short. Otherwise the policy is similar to the one for Scenario 1.1.

b) Scenario 2.1-2.4: IP strategy +#attr selection: The RL policies for jointly optimising IP strategy and attribute selection learn to select the number of attributes according to the generation Scenarios 2.1-2.4. For example, the RL policy learned for Scenario 2.1 generates a RECOMMEND with 5 attributes if the number of database hits is low (<13). Otherwise, it will start with a SUMMARY using 2 attributes. If the user is predicted to narrow down his focus after the SUMMARY, the policy continues with a COMPARE using 1 attribute only, otherwise it helps the user by presenting 4 attributes. It then continues with RECOMMEND(5), and stops as soon as the user is predicted to select one item.

The learned policy for Scenario 2.1 generates 5.85 attributes per NLG turn on average (i.e. the cumulative number of attributes generated in the whole NLG sequence, where the same attribute may be repeated within the sequence). This strategy primarily adapts to the variations from the user simulation (extended bi-gram model). For Scenario 2.2 the average number of attributes is higher (7.15) since the number of attributes helps to narrow down the user's focus via the DBhits/attribute relationship specified in Section V-B. For Scenario 2.3 fewer attributes are generated on average (3.14), since here the number of attributes influences the sentence realiser, i.e. fewer attributes result in fewer sentences, but does not impact the user's focus. In Scenario 2.4 all the conditions mentioned above influence the learned policy. The average number of attributes selected is still low (3.19).

In comparison, the average (cumulative) number of attributes for the WoZ baseline is 7.10. The WoZ baseline generates all the possible IP structures (with 3 or 4 attributes) but is restricted to use only "high-level" features (see Figure 4). By beating this baseline we show the importance of the "lower-level" features. Nevertheless, this wizard policy achieves up to 87.6% of the possible reward on this task,

TABLE IV
TEST RESULTS FOR 1000 DIALOGUES, WHERE *** DENOTES THAT THE RL POLICY IS SIGNIFICANTLY ($p < .001$) BETTER THAN THE BASELINE POLICY, WITH THE STANDARD DEVIATION (±) IS SHOWN IN BRACKETS.

Scenario	Wizard Baseline average Reward	RL average Reward	RL % - Baseline % = % improvement
1.1	-15.82(±15.53)	-9.90***(±15.38)	**89.2%** - 85.6%= 3.6%
1.2	-19.83(±17.59)	-12.83***(±16.88)	**87.4%** - 83.2%= 4.2%
2.1	-12.53(±16.31)	-6.03***(±11.89)	**91.5%** - 87.6%= 3.9%
2.2	-14.15(±16.60)	-14.18(±18.04)	**86.6%** - 86.6%= 0.0%
2.3	-17.43(±15.87)	-9.66***(±14.44)	**89.3%** - 84.6%= 4.7%
2.4	-19.59(±17.75)	-12.78***(±15.83)	**87.4%** - 83.3%= 4.1%

and so can be considered a serious baseline against which to measure performance.

The only case (Scenario 2.2) where RL does not improve significantly over the baseline is where lower level features do not play an important role for learning good strategies: Scenario 2.2 is only sensitive to higher level features (DBhits).

VII. EVALUATION WITH REAL USERS

After obtaining satisfactory results in simulation, we now test whether these results transfer to a real user setting, using a richer set of evaluation metrics. In general, natural language generation for spoken dialogue systems serves two goals: On the one hand the *local* NLG task is to present “enough” information to the user while keeping the utterances short and understandable. In the previous section, we demonstrated that the learned IP model *locally* outperforms the WoZ baseline.

On the other hand, better information presentation should also contribute to the *global/ overall* dialogue task, so as to maximise task completion. In order to test this hypothesis, the learned policy was integrated into a telephone-based spoken dialogue system, and evaluated with real users. In particular, we test its ability to contribute to *overall* dialogue task success.

A. System Integration

In order to evaluate our NLG strategy with real users, it was integrated into the *CamInfo* system [63], a fully statistical spoken dialogue system providing tourist information for real locations in Cambridge. This baseline system has been made accessible by phone using VoIP technology, enabling out-of-lab evaluation with large numbers of users. Apart from practical advantages in managing evaluation campaigns, this development effort was also intended as a step towards evaluating spoken dialogue systems under more realistic conditions. Please note, however, that the users in this evaluation were still recruited and asked to complete predefined tasks (see Section VII-B), and therefore the evaluation might not be as realistic as an evaluation of a final deployed application with real-world users having real goals [64].

The speech recogniser, semantic parser and dialogue manager have all been developed at Cambridge University. For speech synthesis, the Baratinoo synthesiser [9] developed at France Telecom, was used.

The DM uses a POMDP (Partially Observable Markov decision process) framework, allowing it to process N-Best lists of ASR hypotheses and keep track of multiple dialogue state hypotheses. The DM policy is trained to select system dialogue acts given a probability distribution over possible dialogue states. It has been shown that such dialogue managers can exploit the information in the N-Best lists (as opposed to only using the top ASR hypothesis) and are therefore particularly effective in noisy conditions [63].

The natural language generation component of this baseline system is a standard rule-based surface realiser covering the full range of system dialogue acts that the dialogue manager can produce. It has only one IP strategy, i.e., the system only provides information about database entries in the form of single venue recommendations (the RECOMMEND strategy, see Table I). The attributes of the venue to be presented are selected heuristically. This baseline represents the information presentation strategies in conventional slot-filling systems, e.g. [4], [32]. In the extended version of the system, the IP strategy is replaced by our trained NLG component, which is optimised to decide between different IP strategies.

We follow a hybrid between statistical and rule-based approaches in order to integrate our trained IP policy: higher-level hand-coded rules impose a set of constraints on the statistical policy. Note that the possibility of constraining statistical policies with hard-coded rules is increasingly required for developing commercial dialogue systems [65]. We follow a modular approach for integration, where the NLG and dialogue management strategies were trained separately (we discuss this issue further below).

We impose the following rule-based constraints on our policy in order to make it compatible with the (separately trained) DM policy:

- The chosen IP strategy must end with a RECOMMEND action, since the DM expects (exactly one) named entity to be mentioned.
- COMPARE actions are excluded in order to not introduce new named entities that the user may refer to later (since the DM was not optimised under this condition).
- The attribute selection is forced to present at least the attributes chosen by the DM.

The remaining decision points are: choosing between RECOMMEND and SUMMARY+RECOMMEND, as well as selecting additional attributes to present to the user. Although this is a somewhat limited version of the fully optimised IP strategy, it is still interesting to discover whether even a limited amount of NLG optimisation (in terms of more elaborate IP strategies and attribute selection) has an effect on overall global system

TABLE VI
EXAMPLE DIALOGUE WITH THE CAMBRIDGE RESTAURANT INFORMATION DIALOGUE SYSTEM.

Sys1:	How may I help you?
Usr1:	HI I AM LOOKING FOR A MODERATE TURKISH RESTAURANT
Sys2:	I found 2 restaurants, which have Turkish cuisine, and which are in the moderate price range. One of them is located in the Romsey area, and one of them is located in the riverside area. The restaurant called 196 Meze Bar, has the best overall quality amongst the selected restaurants. It is in the moderate price range and located in the Romsey area.
Usr2:	GIVE THE ADDRESS FOR THE RESTAURANT IN THE RIVERSIDE AREA
Sys3:	The restaurant called Efes is a Turkish restaurant in the riverside area. This restaurant is in the moderate price range.
Usr3:	GIVE ME THE PHONE NUMBER
Sys4:	The phone number of Efes is 627 3405
Usr4:	THANK YOU GOODBYE

performance.

Hence, in this real user evaluation, we compared the baseline system, incorporating a single recommendation IP strategy only, with the extended system, incorporating our trained IP policy. In a previous proof-of-concept study [19] a similar rule-based baseline NLG strategy (RECOMMEND only) was shown to be outperformed in simulation. We now evaluate whether these results transfer to real user settings. In the remainder of the paper we will refer to the baseline system as *BASE* system and to the system with the integrated trained IP strategy as *TIP*.

Table VI shows an abbreviated example dialogue from the evaluation corpus. The dialogue contains two IP turns where the system selects a strategy based on the trained IP policy: Sys2 and Sys3. In Sys2, the system decides to first summarise the retrieved restaurants, using *location* as a distinguishing feature, and then recommend the top ranking one. In turn Usr2, the user decides not to select the recommended restaurant, but to ask for a restaurant in a different location, which was previously mentioned in the SUMMARY. Note that even though the system failed to recognise "address" in User2, it did understand "riverside area" and was able to progress the dialogue. This example illustrates one of the benefits of generating summaries for task based dialogue systems: it helps the user to gain an overview of the available options [1].

The dialogue also shows how the system varies between SUMMARY+RECOMMEND (Sys2) and RECOMMEND only (Sys3) based on the progress of the dialogue task, which is reflected in the state space by how many "slots" are filled. In this case, the trained IP policy decides to only perform a RECOMMEND after 3 slots were specified by the user (*cuisine*, *price*, and *area*). SUMMARY, in its function of providing an overview, was chosen at the beginning of the dialogue, when fewer slots were filled. The two following user utterances (Usr3, Usr4) indicate the success of this policy.

B. Experimental Design

Next, we empirically evaluated the two systems using two approaches to recruit and manage subjects. In the first approach, subjects were recruited using mail-shots and web-based advertising amongst people from Cambridge and Edinburgh, mostly students. From the resulting pool of subjects, people were gradually invited to start the tasks, in their own time, and within a given trial period of around two weeks. After the trial period, they were paid (using PayPal) per completed task, with a required minimum of 15 tasks, and a maximum of 40 tasks. For the two systems, this resulted in a corpus of 304 dialogues. In the second approach, an alternative method of managing subjects was used, using Amazon Mechanical Turk [66]. In this setup, tasks are published as so-called HITs (Human Intelligence Tasks) on a web-server and registered workers can complete them. This setup resulted in 532 collected dialogues for the two systems compared[11]. In the remainder of this paper, we will refer to the corpus obtained with 'locally' managed subjects as *LOC* and to the corpus obtained using Amazon Mechanical Turk as *AMT*.

In both of the above-mentioned approaches, the subjects were directed to a webpage with detailed instructions and for each task, a phone number to call and the scenario to follow. The subjects were randomly assigned to interact with one of the systems (BASE or TIP). A scenario describes a place to eat in town, with some constraints, for example: *"You want to find a moderately priced restaurant and it should be in the Riverside area. You want to know the address, phone number, and type of food."* After the dialogue, the subjects were asked to fill in a short questionnaire, assessing the impact of IP strategies on the users' perception of various system components:

Q1. Did you find all the information you were looking for? [Yes / No]

Please state your attitude towards the following statements:

Q2. The system understood me well. [1 – 6]

Q3. The phrasing of the system's responses was good. [1 – 6]

Q4. The system's voice was of good quality. [1 – 6]

1: strongly disagree	4: slightly agree
2: disagree	5: agree
3: slightly disagree	6: strongly agree

Table VII summarises the two corpora of collected data. For the AMT corpus, considerably more subjects were used, although many of them did only a small number of tasks. For the LOC corpus, it was more difficult to recruit many subjects, but in this setup, the subjects could be asked to complete a minimum number of tasks, hence the higher average number of dialogues per user.

Also, note that the word error rate (WER) is relatively high in both corpora. This is partly due to the fact that the ASR module had not been trained specifically for this particular domain due to lack of training data. Furthermore, some of

[11] This evaluation was part of a larger evaluation campaign, in which 2046 dialogues were collected in total. This data is available from the CLASSiC project data repository, see http://www.macs.hw.ac.uk/iLabArchive/CLASSiCProject/Data/myaccount.php

TABLE VII
OVERVIEW OF COLLECTED DATA, WITH FOR EACH CORPUS THE NUMBER OF DIALOGUES (#DIALS), THE AVERAGE NUMBER OF USER TURNS PER DIALOGUE (AVGTURNS), THE NUMBER OF UNIQUE USERS (#USERS), THE AVERAGE NUMBER OF DIALOGUES PER USER (#DSUSR), AND THE WORD ERROR RATE (WER).

Corpus	#Dials	AvgTurns	#Users	#DsUsr	WER
LOC	304	11.48	19	16.00	56.5
AMT	532	10.09	113	4.71	53.6

TABLE VIII
FREQUENCY OF OCCURRENCES OF EACH IP STRATEGY OBSERVED IN THE EVALUATION WITH NUMBER OF ATTRIBUTES IN BRACKETS.

Frequ.	Strategy(attributes)
1	RECOMMEND(1)
123	RECOMMEND(2)
163	RECOMMEND(3)
254	SUMMARY(1)+RECOMMEND(1)
778	SUMMARY(2)+RECOMMEND(2)
270	SUMMARY(3)+RECOMMEND(3)

TABLE X
SUBJECTIVE EVALUATION RESULTS, BASED ON THE QUESTIONNAIRE [Q1-Q4], WHERE AN ASTERISK (*) DENOTES A SIGNIFICANT DIFFERENCE AT $p < 0.05$ (USING A Z-TEST FOR Q1 AND A MANN-WHITNEY TEST FOR Q2–Q4).

Corpus	System	Q1	Q2	Q3	Q4
LOC	BASE	65.33	3.69	3.94	**4.23***
	TIP	60.00	3.44	3.70	3.91
AMT	BASE	64.18	3.92	4.16	3.81
	TIP	56.15	3.87	4.30	3.85
TOT	BASE	64.56	3.85	4.10	3.95
	TIP	57.87	3.68	4.03	3.88

the subjects were non-native speakers and some subjects used Skype to call the systems, which causes distortion of the audio signal. These conditions are the same for both BASE and TIP systems. Despite the high ASR error rates, overall task completion rates were relatively high [67].

The overall most frequently employed IP strategy is SUMMARY(2)+RECOMMEND(2), see Table VIII. Also, note that the trained policy never employed more than 3 attributes, and always chose to use the same number of attributes for its combined IP strategies.

C. Results

After processing the log files and completed user questionnaires, both objective and subjective performance measures were computed in order to compare the systems.

1) Objective evaluation: For the objective evaluation of the two dialogue systems we focused on measuring goal completion rates, which can be done in different ways. First, we can take the task specification assigned to the user for each dialogue and then analyze the system dialogue acts. *Partial completion* (ObjSucc-PC) is achieved when the system has offered a venue that matches the constraints as specified in the assigned goal, for example it has provided the name of a cheap chinese restaurant in the riverside area. *Full completion* (ObSucc-FC) is achieved when the system has also provided the required additional information about that venue, for example the phone number and address.

Table IX shows all success rates obtained from the evaluation, for the corpus with data from locally recruited subjects (LOC), and the corpus with data from Amazon Mechanical Turk (AMT) workers, as well as both corpora pooled together (TOT). The results show that the system with our NLG component (TIP) outperforms the baseline system (BASE) on all objective success rates in both corpora. Despite an overall low completion rate (which as mainly due to a high WER rate as explained earlier), relative improvements of up to 30% for full completion on the AMT corpus were obtained. After pooling the two corpora together, we have a sufficient number of dialogues to show that the improvement from our NLG strategy is statistically significant on both partial and full completion (using a 2-tailed z-test for two proportions).

It is also interesting to note that the average number of user turns per dialogue is not significantly different between systems in both corpora, suggesting that the contribution of the trained IP policy to system performance manifests itself primarily in terms of effectiveness rather than efficiency. By providing more useful information to the user, the system might help them to find an appropriate venue in fewer turns, but due to the more lengthy system prompts, more turns might be needed to recover from any speech recognition errors (see WER in Table VII).

2) Subjective evaluation: Table X summarises the subjective user scores from the questionnaire (see Section VII-B). In terms of subjective success rates (Q1), the baseline system (BASE) obtains slightly higher scores on both corpora, although no statistically significant differences were found. We will discuss these results further in Section VII-D.

When comparing the other subjective scores (Q2–Q4) on a scale of [1–6], using a Mann-Whitney test, the only case where a statistically significant difference is found between the two systems is the score for *Q4:VoiceQuality* in the LOC corpus, where the baseline system is significantly better. Since the TTS voice is exactly the same for both systems, the difference in perceived voice quality might be influenced by the longer system prompts for the TIP system. However, we did not observe this pattern in the AMT corpus.

We also compared the Mechanical Turk setup to the setup where subjects where recruited locally (AMT vs. LOC for both systems). For the TIP system, *Q2:Understanding* and *Q3:Phrasing* are significantly higher in the AMT corpus compared to the LOC corpus. Similarly, the BASE system performs significantly better for *Q3:Phrasing* under the Mechanical Turk setting. However, when combining the results for all the subjective scores (similar to the objective scores), none of the differences are significant.

In sum, there is no difference in user ratings between the original BASE system and the TIP system with the integrated trained NLG strategy, except for *Q4:VoiceQuality*, which is better rated for the BASE system in the LOC corpus, even though the systems had identical TTS. The difference in

TABLE IX

Overview of all success rates (%) obtained for the two corpora, including subjective success obtained from Q1 of the user questionnaire(SubjSucc), objective success based on assigned goals (ObjSucc-PC for partial completion and ObjSucc-FC for full completion). 95% confidence intervals for all success rates are indicated in brackets; statistically significant improvements ($p < 0.05$ using a z-test) are indicated with an asterisk (*).

Corpus	System	#Dials	#Turns	SubjSucc	ObjSucc-PC	ObjSucc-FC
LOC	BASE	199	11.69	65.33 (6.61)	73.37 (6.14)	46.73 (6.93)
	TIP	105	11.02	60.00 (9.37)	77.23 (8.02)	49.50 (9.56)
AMT	BASE	402	9.86	64.18 (4.69)	51.00 (4.89)	28.86 (4.43)
	TIP	130	10.83	56.15 (8.53)	60.77 (8.39)	37.69 (8.33)
TOT	BASE	601	10.46	64.56 (3.82)	58.40 (3.94)	34.78 (3.81)
	TIP	235	10.91	57.87 (6.31)	$\mathbf{68.09}$ $(\mathbf{5.96})*$	$\mathbf{42.98}$ $(\mathbf{6.33})*$

ratings between the LOC and AMT corpora suggests that the way in which subjects are recruited, instructed and paid, as well as the user population targeted, has an impact on subjective ratings obtained.

D. Discussion of Results

The results showed that the trained information presentation model significantly improves objective dialogue task completion, with up to a 23% increase (+8.2% raw improvement) compared to hand-coded presentation prompts often used in conventional slot-filling dialogue systems.

It also shows that the choice between generating a SUMMARY or generating a RECOMMEND only, as well as a principled way for selecting attributes, has a significant effect on task success.

The subjective scores however were quite similar between the two systems, and in terms of perceived success rate, the baseline system scored slightly better, though not statistically significantly.

An important factor that may have influenced the results, was that the word error rate was relatively high throughout the data. The more elaborate information presentation prompts from the integrated system (TIP) might have exacerbated the many speech recognition problems, where the DM might have falsely initiated a lengthy information presentation prompt after a mis-recognition error. This is also suggested by the analysis of dialogue length, which turned out to be very similar between the two systems. By providing more useful information to the user, the TIP system might help them to find an appropriate venue in fewer turns, but due to the lengthy system prompts, more turns might be needed to recover from speech recognition errors.

Also note that the results only show significant improvements for objective task success metrics, even though the NLG strategy was optimised for user satisfaction (using a linear regression model for estimation, see Section V-C). One possible explanation for this is the presence of ASR errors in the real user evaluation, which can negatively impact user satisfaction. So even though the NLG was optimised for user satisfaction, this may have been over-ridden by ASR errors. A logistic regression analysis showed a strong correlation between WER and subjective task success [67].

Although these evaluation results are positive, a system setup which combines separately trained dialogue manager and NLG components is not ideal. In this case the dialogue manager was trained in a setup where only the single item recommendation strategy for IP is used. Therefore, for the dialogue manager state update, only dialogue acts for such IP prompts are expected. If the trained NLG model decides to use an alternative IP strategy, a mismatch is then potentially caused between what the dialogue manager planned and what is actually presented to the real user. Therefore, the NLG module might result in user behaviour that the dialogue manager is not optimised for. As a practical compromise it was therefore decided (as explained above) to require all IP prompts to end with a single item recommendation, and the COMPARE strategy was blocked during the evaluation. Therefore, neither DM nor NLG were trained for the final operating conditions that they would experience in this application, though the constraints on NLG mentioned above meant that the DM's chosen actions were maintained. In future work we therefore plan to jointly optimise the DM and NLG strategies (see also [68]), and it is likely that full use of an optimised IP strategy would lead to an even greater performance boost in the overall system. We expect that a joint optimisation of DM and NLG policies would prevent the DM from initiating long IP prompts after likely mis-recognitions. We predict that the results obtained in this study would be even stronger for a jointly-optimised DM+NLG strategy, and we pursue this in current work.

Finally we note that in the above user evaluation the fact that users were assigned a task may in some cases have limited the beneficial effects of providing a summary. The benefits might be greater if users have their own genuine goals, or more specifically, develop their goals during the dialogue based on possible attributes that the system mentions. For example, when the system gives a summary using different food types, the user may then be inclined to decide which food type they want. The benefits of summaries may therefore be greater when subjects are free to choose their own goals while browsing.

VIII. Conclusion and Future Work

In this paper we present and evaluate a novel framework for adaptive natural language generation where the problem is formulated as stochastic incremental planning under uncertainty, which can be approached using reinforcement learning methods. At the time of writing, this approach is being actively explored by a variety of researchers [23], [22], [7], [24], [6], [21], [13], [26], [25], and there is closely related work which also explores NLG as a process of maximising utility using Game Theory [46], [47].

We apply this framework to adaptive information presentation (IP) in spoken dialogue systems. For the IP problem a statistical optimisation framework was developed for content structure planning and attribute selection. This work was the first to apply a data-driven optimisation method to this decision space – from data collection to user testing. The IP model is adaptive to variability observed in a stochastic SDS, and it incrementally adapts the IP policy at the turn level. Reinforcement learning is used to automatically optimise the IP policy with respect to a data-driven objective function. An evaluation found that the trained information presentation strategy significantly improves dialogue task completion for real users, with up to a 8.2% increase (23% relative) compared to a deployed dialogue system which uses conventional, hand-coded presentation prompts.

This methodology provides new insights into the nature of the IP problem, which has previously been treated as a sequential module following dialogue management with no access to lower-level context features. Our results suggest that features from modules which traditionally follow NLG strategy and attribute selection decisions, e.g. features from surface realisation or predicted TTS quality, play an important role for IP policy optimisation. As such, we argue that the traditional pipeline approach for SDS is not appropriate for SDS that include stochastic modules.

It is also interesting to note that all the user studies show that an adaptive NLG component significantly contributes to the (perceived or objective) task success of the system. Thus, such data-driven adaptive NLG strategies have "global" effects on overall system performance. The data-driven planning methods applied here therefore promise significantly upgraded performance of generation modules, and thereby of Natural Language interaction in general.

A. Future work

There are several directions in which this research can be developed. An interesting challenge for NLG in general, is that of 'generation under uncertainty' [69], [70], [71], where language must be generated for users even though there is some uncertainty about their state. This uncertainty can be about their location, their gaze direction and objects in their field of view, or even about their goals and preferences. Regarding generation under uncertainty, an interesting research direction will be to explicitly represent uncertainty about the generation context using techniques such as belief states in Partially Observable Markov decision processes (POMDPs).

We have currently only investigated a small number of "lower level" features, such as sentence length. Future work could also include the predicted TTS quality [9] as a feature for optimising NLG decisions.

In addition, the issue of incremental NLG in spoken dialogue systems must be tackled for more natural and efficient dialogue systems [72], [73]. Here, phenomena such as split utterances and barge-in are a research focus, which we are currently targeting using statistical approaches to NLG in the EC FP7 PARLANCE project (www.parlance-project.eu) [13], [26].

Finally, in the ERC "Strategic Conversation" (STAC) project [74], [75] we currently investigate how this methodology scales to more complex generation scenarios, in particular to multi-agent adversarial, non-collaborative, settings where NLG utterances contain complex rhetorical relations and can be planned so as to hide information or mislead hearers.

Note that all data sets collected in this work are available from the CLASSiC project data repository [76].[12]

Acknowledgment

We would like to thank Xingkun Liu, Helen Hastie, and Paul Crook for their collaboration on the data collections.

The research leading to this work has received funding from the European Community's FP7 programmes: (FP7/2007-13) under grant agreement no. 216594 (CLASSiC); (FP7/2011-14) under grant agreement no. 270019 (SPACEBOOK); (FP7/2011-16) under grant agreement no. 269427 (STAC); and partially funded by the Engineering and Physical Sciences Research Council, UK (EPSRC) under project no. EP/G069840/1

References

[1] J. Polifroni and M. Walker, "Intensional Summaries as Cooperative Responses in Dialogue Automation and Evaluation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2008.

[2] V. Demberg and J. Moore, "Information presentation in spoken dialogue systems," in *Proc. of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2006.

[3] M. Walker, S. Whittaker, A. Stent, P. Maloor, J. Moore, M. Johnston, and G. Vasireddy, "User tailored generation in the MATCH multimodal dialogue system," *Cognitive Science*, vol. 28, pp. 81–840, 2004.

[4] S. Young, J. Schatzmann, K. Weilhammer, and H. Ye, "The Hidden Information State Approach to Dialog Management," in *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2007.

[5] V. Demberg, A. Winterboer, and J. D. Moore, "A strategy for information presentation in spoken dialog systems," *Computational Linguistics*, vol. 37, no. 3, pp. 489—539, 2011.

[6] S. Janarthanam and O. Lemon, "Adaptive referring expression generation in spoken dialogue systems: Evaluation with real users," in *Proc. of the Annual SIGDIAL Conference on Discourse and Dialogue*, 2010.

[7] S. Janarthanam, H. Hastie, O. Lemon, and X. Liu, "'The day after the day after tomorrow?' A machine learning approach to adaptive temporal expression generation: training and evaluation with real users," in *Proc. of the Annual SIGDIAL Conference on Discourse and Dialogue*, 2011.

[8] C. Nakatsu and M. White, "Learning to say it well: Reranking realizations by predicted synthesis quality," in *Proc. of the 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL)*, 2006.

[9] C. Boidin, V. Rieser, L. van der Plas, O. Lemon, and J. Chevelu, "Predicting how it sounds: Re-ranking alternative inputs to TTS using latent variables," in *Proc. of Interspeech/ICSLP, Special Session on Machine Learning for Adaptivity in Spoken Dialogue Systems*, 2009.

[10] S. Young, "Probabilistic methods in spoken dialogue systems," *Philosophical Trans Royal Society (Series A)*, vol. 358, no. 1769, pp. 1389–1402, 2000.

[11] G. Skantze and D. Schlangen, "Incremental dialogue processing in a micro-domain," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, ser. EACL '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 745–753. [Online]. Available: http://dl.acm.org/citation.cfm?id=1609067.1609150

[12] O. Lemon and O. Pietquin, Eds., *Data-Driven Methods for Adaptive Spoken Dialogue Systems Computational Learning for Conversational Interfaces*. Springer, 2012.

[12] http://www.macs.hw.ac.uk/iLabArchive/CLASSiCProject/Data/myaccount.php

[13] N. Dethlefs, H. Hastie, V. Rieser, and O. Lemon, "Optimising incremental generation for spoken dialogue systems: Reducing the need for fillers by optimising waiting time and content re-ordering," in *Proc. of the 7th International Conference on Natural Language Generation (INLG)*, 2012.

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

[15] J. Williams and S. Young, "Partially Observable Markov Decision Processes for spoken dialog systems," *Computer Speech and Language*, vol. 21, no. 2, pp. 231–422, 2007.

[16] J. Henderson, O. Lemon, and K. Georgila, "Hybrid Reinforcement / Supervised Learning of Dialogue Policies from Fixed Datasets," *Computational Linguistics*, vol. 34, no. 4, pp. 487 – 513, 2008.

[17] V. Rieser and O. Lemon, "Learning and evaluation of dialogue strategies for new applications: Empirical methods for optimization from small data sets," *Computational Linguistics*, vol. 37, no. 1, pp. 153–196, 2011.

[18] O. Lemon, "Adaptive natural language generation in dialogue using Reinforcement Learning," in *Proc. of the 12th SEMdial Workshop on the Semantics and Pragmatics of Dialogues*, London, UK, June 2008.

[19] V. Rieser and O. Lemon, "Natural Language Generation as Planning Under Uncertainty for Spoken Dialogue Systems," in *Proc. of the Conference of European Chapter of the Association for Computational Linguistics (EACL)*, 2009.

[20] V. Rieser, O. Lemon, and X. Liu, "Optimising Information Presentation for Spoken Dialogue Systems," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Uppsala, Sweden, July 2010, pp. 1009–1018.

[21] V. Rieser, S. Keizer, O. Lemon, and X. Liu, "Adaptive Information Presentation for Spoken Dialogue Systems: Evaluation with human subjects," in *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG)*, 2011.

[22] N. Dethlefs, H. Cuayáhuitl, and J. Viethen, "Optimising natural language generation decision making for situated dialogue," in *Proc. of the Annual SIGDIAL Conference on Discourse and Dialogue*, 2011.

[23] N. Dethlefs and H. Cuayáhuitl, "Hierarchical reinforcement learning and hidden markov models for task-oriented natural language generation," in *Proc. of 49th Annual Meeting of the Association for Computational Linguistics*, 2011.

[24] ——, "Combining hierarchical reinforcement learning and bayesian networks for natural language generation in situated dialogue," in *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG)*, 2011.

[25] N. Bertomeu, "Finding optimal presentation sequences for a conversational recommender system," in *Advances in Computational Intelligence*, ser. Communications in Computer and Information Science, S. Greco, B. Bouchon-Meunier, G. Coletti, M. Fedrizzi, B. Matarazzo, and R. Yager, Eds. Springer Berlin Heidelberg, 2012, vol. 300, pp. 328–337. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-31724-8_34

[26] N. Dethlefs, H. Hastie, V. Rieser, and O. Lemon, "Optimising incremental dialogue decisions using information density for interactive systems," in *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing*, 2012.

[27] M. Walker, R. Passonneau, and J. Boland, "Quantitative and qualitative evaluation of DARPA Communicator spoken dialogue systems," in *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2001.

[28] K. Dohsaka, N. Yasuda, and K. Aikawa, "Efficient spoken dialogue control depending on the speech recognition rate and system's database," in *Proc. Interspeech*, 2003.

[29] T. Misu and T. Kawahara, "Bayes risk-based dialogue management for document retrieval system with speech interface," *Speech Communication*, vol. 52, no. 1, pp. 61–71, 2010.

[30] M. Walker, A. Stent, F. Mairesse, and R. Prasad, "Individual and domain adaptation in sentence planning for dialogue," *Journal of Artificial Intelligence Research (JAIR)*, vol. 30, pp. 413–456, 2007.

[31] C. Nakatsu, "Learning contrastive connectives in sentence realization ranking," in *Proc. of SIGdial Workshop on Discourse and Dialogue*, 2008.

[32] B. Thomson and S. Young, "Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems," *Computer Speech and Language*, vol. 24, no. 4, pp. 562–588, 2010.

[33] S. Whittaker, M. Walker, J. Moore, S. Whittaker, M. Walker, and J. Moore, "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)*, 2002.

[34] J. Polifroni and M. Walker, "Learning database content for spoken dialogue system design," in *Proc. of the IEEE/ACL workshop on Spoken Language Technology (SLT)*, 2006.

[35] G. Carenini and J. D. Moore, "Generating and evaluating evaluative arguments," *Artificial Intelligence*, vol. 170, no. 11, pp. 925–952, 2006.

[36] M. Walker, S. Whittaker, A. Stent, P. Maloor, J. Moore, M. Johnston, and G. Vasireddy, "Generation and evaluation of user tailored responses in multimodal dialogue." *Cognitive Science*, vol. 28, no. 5, pp. 811–840, 2004.

[37] A. Koller and R. Petrick, "Experiences with planning for natural language generation," in *Proc. of ICAPS*, 2008.

[38] V. Rieser and O. Lemon, "Learning effective multimodal dialogue strategies from Wizard-of-Oz data: Bootstrapping and evaluation," in *Proc. of the 21st International Conference on Computational Linguistics and 46th Annual Meeting of the Association for Computational Linguistics (ACL/HLT)*, Columbus, Ohio, USA, June 2008, pp. 638–646.

[39] A. Stent, R. Prasad, and M. Walker, "Trainable sentence planning for complex information presentation in spoken dialog systems," in *Proceedings of Association for Computational Linguistics (ACL)*, 2004.

[40] A. Stent, M. Walker, S. Whittaker, and P. Maloor, "User-tailored generation for spoken dialogue: an experiment," in *Proc. of International Conference on Spoken Language Processing (ICSLP)*, 2002.

[41] S. Whittaker, M. Walker, and P. Maloor, "Should i tell all? an experiment on conciseness in spoken dialogue," in *Proc. European Conference on Speech Processing (EUROSPEECH)*, 2003.

[42] J. Moore, M. E. Foster, O. Lemon, and M. White, "Generating tailored, comparative descriptions in spoken dialogue," in *Proc. FLAIRS*, 2004.

[43] A. H. Oh and A. I. Rudnicky, "Stochastic natural language generation for spoken dialog systems," *Computer Speech & Language*, vol. 16, no. 3/4, pp. 387—407, 2002.

[44] F. Mairesse and M. A. Walker, "Controlling user perceptions of linguistic style: Trainable generation of personality traits," *Computational Linguistics*, vol. 37, no. 3, pp. 455–488, 2012/05/16 2011. [Online]. Available: http://dx.doi.org/10.1162/COLI_a_00063

[45] R. Sutton and A. Barto, *Reinforcement Learning*. MIT Press, 1998.

[46] K. van Deemter, "Utility and Language Generation: The case of Vagueness," *Journal of Philosophical Logic*, pp. 607–632, 2009.

[47] D. Golland, P. Liang, and D. Klein, "A game-theoretic approach to generating spatial descriptions,," in *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2010.

[48] P. A. Crook and O. Lemon, "Lossless Value Directed Compression of Complex User Goal States for Statistical Spoken Dialogue Systems," in *Proceedings of Interspeech*, 2011.

[49] V. Rieser, I. Kruijff-Korbayová, and O. Lemon, "A corpus collection and annotation framework for learning multimodal clarification strategies," in *Proc. of the 6th SIGdial Workshop on Discourse and Dialogue*, Lisbon, Portugal, September 2005, pp. 97–106.

[50] V. Rieser, X. Liu, and O. Lemon, "Optimal Wizard NLG Behaviours in Context," Deliverable 4.2, CLASSiC Project, Tech. Rep., 2009.

[51] X. Liu, V. Rieser, and O. Lemon, "A wizard-of-oz interface to study information presentation strategies for spoken dialogue systems," in *Proc. of the 1st International Workshop on Spoken Dialogue Systems*, 2009.

[52] W. W. Cohen, "Fast effective rule induction," in *Proceedings of the 12th International Conference on Machine Learning (ICML)*, Tahoe City, California, USA, July 1995, pp. 115–123.

[53] V. Rieser and O. Lemon, *Reinforcement Learning for Adaptive Dialogue Systems: A Data-driven Methodology for Dialogue Management and Natural Language Generation*, ser. Theory and Applications of Natural Language Processing, G. Hirst, E. Hovy, and M. Johnson, Eds. Springer, 2011.

[54] J. Henderson and O. Lemon, "Mixture Model POMDPs for Efficient Handling of Uncertainty in Dialogue Management," in *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2008.

[55] S. Keizer, M. Gasic, F. Mairesse, B. Thomson, K. Yu, and S. J. Young, "Modelling user behaviour in the his-pomdp dialogue manager," in *Proc. of the IEEE/ACL Spoken Language Technology (SLT)*, 2008.

[56] W. Eckert, E. Levin, and R. Pieraccini, "User modeling for spoken dialogue system evaluation," in *Proc. of the IEEE workshop on Automatic Speech Recognition and Understanding (ASRU)*, Santa Barbara, CA, USA, December 1997, pp. 80–87.

[57] P. Clarkson and R. Rosenfeld, "Statistical Language Modeling Using the CMU-Cambridge Toolkit," in *Proc. of ESCA Eurospeech*, 1997.

[58] H. Cuayáhuitl, S. Renals, O. Lemon, and H. Shimodaira, "Human-Computer Dialogue Simulation Using Hidden Markov Models," in *Proc. of the IEEE workshop on Automatic Speech Recognition and*

Understanding (ASRU), San Juan, Puerto Rico, November 2005, pp. 290–295.

[59] S. Jung, C. Lee, K. Kim, M. Jeong, and G. G. Lee, "Data-driven user simulation for automated evaluation of spoken dialog systems," *Computer Speech & Language*, vol. 23, pp. 479–509, 2009.

[60] S. Janarthanam and O. Lemon, "A Two-tier User Simulation Model for Reinforcement Learning of Adaptive Referring Expression Generation Policies," in *Proc. of the Annual SIGdial Conference on Discourse and Dialogue*, 2009.

[61] M. Walker, C. Kamm, and D. Litman, "Towards developing general models of usability with PARADISE," *Natural Language Engineering*, vol. 6, no. 3, pp. 363–377, 2000.

[62] D. Shapiro and P. Langley, "Separating skills from preference: Using learning to program by reward," in *Proc. of the 19th International Conference on Machine Learning (ICML)*, Sydney, Australia, July 2002, pp. 570–577.

[63] S. Young, M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu, "The Hidden Information State Model: a practical framework for POMDP-based spoken dialogue management," *Computer Speech and Language*, vol. 24, no. 2, pp. 150–174, 2010.

[64] A. Black, S. Burger, A. Conkie, H. Hastie, S. Keizer, O. Lemon, N. Merigaud, G. Parent, G. Schubiner, B. Thomson, J. D. Williams, K. Yu, S. Young, and M. Eskenazi, "Spoken Dialog Challenge 2010: Comparison of Live and Control Test Results," in *Proc. of SIGdial Conference on Discourse and Dialogue*, 2011.

[65] J. Williams, "The best of both worlds: Unifying conventional dialog systems and POMDPs," in *Proceedings of Interspeech*, 2008.

[66] F. Jurcicek, S. Keizer, M. Gasic, F. Mairesse, B. Thomson, K. Yu, and S. Young, "Real user evaluation of spoken dialogue systems using amazon mechanical turk," in *Proc. Interspeech*, Florence, Italy, August 2011.

[67] R. Laroche, G. Putois, P. Bretier, M. Aranguren, J. Velkovska, H. Hastie, S. Keizer, K. Yu, F. Jurcicek, O. Lemon, and S. Young, "D6.4 Final evaluation of CLASSiC TownInfo and Appointment Scheduling systems," The CLASSIC Project (FP7/2008-2011 grant agreement no. 216594), Tech. Rep., 2011.

[68] O. Lemon, "Learning what to say and how to say it: joint optimization of spoken dialogue management and Natural Language Generation," *Computer Speech and Language*, vol. 25, no. 2, pp. 210–221, 2011.

[69] O. Lemon, S. Janarthanam, and V. Rieser, "Generation under uncertainty," in *Proceedings of INLG / Generation Challenges*, 2010.

[70] S. Janarthanam and O. Lemon, "The GRUVE Challenge: Generating Routes under Uncertainty in Virtual Environments," in *Proceedings of ENLG / Generation Challenges*, 2011.

[71] S. Janarthanam, X. Liu, and O. Lemon, "A web-based evaluation framework for spatial instruction-giving systems," in *Proc. of Annual Meeting of the Association for Computational Linguistics (ACL)*, 2012.

[72] G. Skantze and A. Hjalmarsson, "Towards Incremental Speech Generation in Dialogue Systems," in *Proceedings of the 11th Annual SigDial Meeting on Discourse and Dialogue*, Tokyo, Japan, 2010.

[73] D. Schlangen and G. Skantze, "A General, Abstract Model of Incremental Dialogue Processing," *Dialogue and Discourse*, vol. 2(1), 2011.

[74] N. Asher, A. Lascarides, O. Lemon, M. Guhe, V. Rieser, P. Muller, S. Afantenos, F. Benamara, L. Vieu, P. Denis, S. Paul, S. Keizer, and C. Degremont, "Modelling strategic conversation: the STAC project," in *The 16th workshop on the semantics and Pragmatics of Dialogue (SeineDial'12)*, 2012.

[75] V. Rieser, O. Lemon, and S. Keizer, "Opponent modelling for optimising strategic dialogue," in *The 16th workshop on the semantics and Pragmatics of Dialogue (SeineDial'12)*, 2012.

[76] V. Rieser and O. Lemon, "Developing dialogue managers from limited amounts of data," in *Data-Driven Methods for Adaptive Spoken Dialogue Systems Computational Learning for Conversational Interfaces*, O. Lemon and O. Pietquin, Eds. Springer, 2012.

Verena Rieser is a Lecturer in Computer Science at Heriot-Watt University, Edinburgh. Her interests include Spoken Dialog Systems, planning under uncertainty, and Natural Language Generation. She recently published a monograph in this area. She is on the Scientific Advisory Committee of SigDial (the Special Interest Group on Dialog and Discourse) as well as on the Scientific Committee of SigGen (the Special Interest Group on Natural Language Generation). In 2014, she serves as area chair for discourse and dialogue for the European Chapter of the Association for Computational Linguistics (EACL). She received her PhD (summa cum laude) from Saarland University, Germany, in 2008, winning the Eduard-Martin prize. Prior to her lectureship, she conducted postdoctoral research at the University of Edinburgh.

Oliver Lemon leads the Interaction Lab in the School of Mathematical and Computer Sciences (MACS) at Heriot-Watt University, Edinburgh. He previously worked at the School of Informatics, University of Edinburgh, and at Stanford University. His main expertise is in the area of machine learning methods for intelligent and adaptive multimodal interfaces, including work on Speech Recognition, Spoken Language Understanding, Dialogue Management, and Natural Language Generation. He applies this work in new interfaces for mobile search, virtual characters, Technology Enhanced Learning, and Human-Robot Interaction, in a variety of international research projects.

Simon Keizer is a Research Fellow in the Interaction Lab in the School of Mathematical and Computer Sciences (MACS) at Heriot-Watt University, Edinburgh (UK). He has an Msc. in Applied Mathematics and a PhD in Computer Science, both obtained at the University of Twente (NL). The main focus of his research is on interaction management and user simulation for training and evaluating spoken dialogue systems and in particular the use of machine learning techniques in this area. As a Research Associate at Tilburg University (NL) and Cambridge University (UK), he has been involved in both national and international collaborative research projects. In the Interaction Lab, he currently focuses on machine learning techniques for multi-user social human robot interaction.