Optimising Information Presentation for Spoken Dialogue Systems

Verena Rieser
University of Edinburgh
Edinburgh, United Kingdom
verena.rieser@ed.ac.uk

Oliver Lemon
Heriot-Watt University
Edinburgh, United Kingdom
o.lemon@hw.ac.uk

Xingkun Liu
Heriot-Watt University
Edinburgh, United Kingdom
x.liu@hw.ac.uk

Abstract

We present a novel approach to Information Presentation (IP) in Spoken Dialogue Systems (SDS) using a data-driven statistical optimisation framework for content planning and attribute selection. First we collect data in a Wizard-of-Oz (WoZ) experiment and use it to build a supervised model of human behaviour. This forms a baseline for measuring the performance of optimised policies, developed from this data using Reinforcement Learning (RL) methods. We show that the optimised policies significantly outperform the baselines in a variety of generation scenarios: while the supervised model is able to attain up to 87.6% of the possible reward on this task, the RL policies are significantly better in 5 out of 6 scenarios, gaining up to 91.5% of the total possible reward. The RL policies perform especially well in more complex scenarios. We are also the first to show that adding predictive "lower level" features (e.g. from the NLG realiser) is important for optimising IP strategies according to user preferences. This provides new insights into the nature of the IP problem for SDS.

1 Introduction

Work on evaluating SDS suggests that the Information Presentation (IP) phase is the primary contributor to dialogue duration (Walker et al., 2001), 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.

In the following we show that IP for SDS can be treated as a data-driven joint optimisation problem, and that this outperforms a supervised model of human 'wizard' behaviour on a particular IP task (presenting sets of search results to a user).

A similar approach has been applied to the problem of Referring Expression Generation in dialogue (Janarthanam and Lemon, 2010).

1.1 Previous work on Information Presentation in SDS

Broadly speaking, IP for SDS can be divided into two main steps: 1) IP strategy selection and 2) Content or Attribute Selection. Prior work has presented a variety of **IP strategies** for structuring information (see examples in Table 1). For example, the SUMMARY strategy is used to guide the user's "focus of attention". It draws the user's attention to relevant attributes by grouping the current results from the database into clusters, e.g. (Polifroni and Walker, 2008; Demberg and Moore, 2006). Other studies investigate a COMPARE strategy, e.g. (Walker et al., 2007; Nakatsu, 2008), while most work in SDS uses a RECOMMEND strategy, e.g. (Young et al., 2007). In a previous proof-of-concept study (Rieser and Lemon, 2009) we show 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 combinations of the strategies, for example SUMMARY followed by RECOMMEND, e.g. (Whittaker et al., 2002), see Figure 1.

Prior work on **Content or Attribute Selection** has used a "Summarize and Refine" approach (Polifroni and Walker, 2008; Polifroni and Walker, 2006; Chung, 2004). This method employs utility-based attribute selection with respect to how each attribute (e.g. price or food type in restaurant

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1009–1018,
Uppsala, Sweden, 11-16 July 2010.

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 (Demberg and Moore, 2006; Winterboer et al., 2007). Our data collection (see Section 3) and training environment 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 attribute selection), and to show the utility of both lower-level features (e.g. from the NLG realiser) and higher-level features (e.g. from Dialogue Management) for this problem. Previous work has only 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. behaviour of the realiser). In Section 4.3 we show that lower level features significantly influence users' ratings of IP strategies. In the following we use a Reinforcement Learning (RL) as a statistical planning framework (Sutton and Barto, 1998) to explore the contextual features for making these decisions, and propose a new joint optimisation method for IP strategies combining content structuring and attribute selection.

2 NLG as planning under uncertainty

We follow the overall framework of NLG as planning under uncertainty (Lemon, 2008; Rieser and Lemon, 2009; Lemon, 2010), where each NLG action is a sequential decision point, based on the current dialogue context and the expected long-term utility or "reward" of the action. Other recent approaches describe this task as planning, e.g. (Koller and Petrick, 2008), or as contextual decision making according to a cost function (van Deemter, 2009), but not as a statistical planning problem, where uncertainty in the stochastic environment is explicitly modelled. Below, we apply this framework to Information Presentation strategies in SDS using Reinforcement Learning, where the example task is to present a set of search results (e.g. restaurants) to users. In particular, we consider 7 possible policies for structuring the content (see Figure 1): Recommending one single item, comparing two items, summarising all of them, or ordered combinations of those actions, e.g. first summarise all the retrieved items and then recommend one of them. The IP module has to decide which action to take next, how many attributes to mention, and when to stop generating.

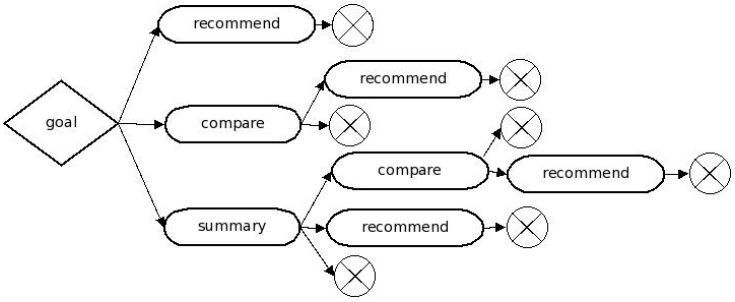

Figure 1: Possible Information Presentation structures (X=stop generation)

3 Wizard-of-Oz data collection

In an initial Wizard-of-Oz (WoZ) study, we asked humans (our "wizards") to produce good IP actions in different dialogue contexts, when interacting in spoken dialogues with other humans (the "users"), who believed that they were talking to an automated SDS. The wizards were experienced researchers in SDS and were familiar with the search domain (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. They also received prior training on this task.

The task for the wizards was to decide which IP structure to use next (see Section 3.2 for a list of IP strategies to choose from), which attributes to mention (e.g. cuisine, price range, location, food quality, and/or service quality), and whether to stop generating, given varying numbers of database matches, varying prompt realisations, and varying user behaviour. Wizard utterances were synthesised using a state-of-the-art text-to-speech engine. The user speech input was delivered to the wizard using Voice Over IP. Figure 2 shows the web-based interface for the wizard.

3.1 Experimental Setup and Data collection

We collected 213 dialogues with 18 subjects and 2 wizards (Liu et al., 2009). Each user performed a total of 12 tasks, where no task set was seen twice by any one wizard. 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 prior experience of

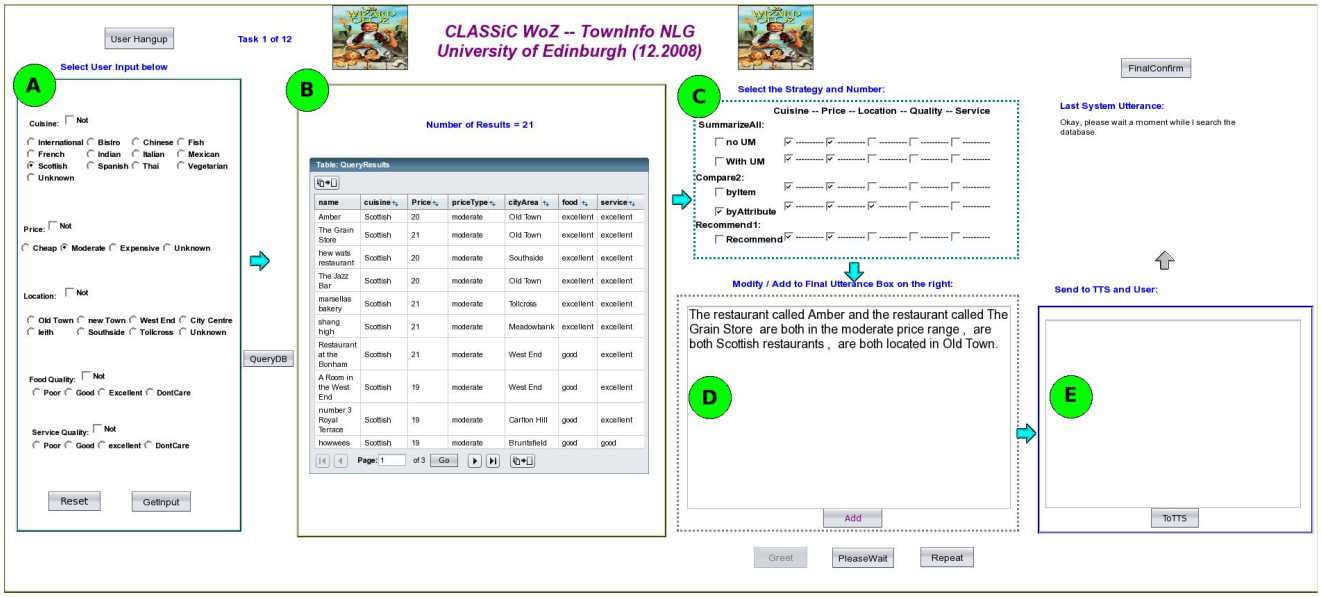

Figure 2: Wizard interface. [A:] The wizard selects attribute values as specified by the user's query. [B:] The retrieved database items are presented in an ordered list. We use a User Modelling approach for ranking the restaurants, see e.g. (Polifroni and Walker, 2008). [C:] The wizard then chooses 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 realiser 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 or to start over again. The text in the final panel is sent to the user via TTS, once the wizard decides to stop generating.

Strategy	Example utterance
SUMMARY no UM	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 UM	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 Saffrani, which is also an Indian restaurant, is in the moderate price range.
COMPARE by Attribute	The restaurant called Kebab Mahal and the restaurant called Saffrani are both Indian restaurants. However, Kebab Mahal is in the cheap price range while Saffrani 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.

Table 1: 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.

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. 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)[1] from the XML logfiles after each dialogue. Please see (Rieser et al., 2009) for more details.

[1]The full corpus and list of features is available at `https://www.classic-project.org/corpora/`

3.2 NLG Realiser

In the Wizard-of-Oz environment 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 (Stent et al., 2004). 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 dynamic sentence generation. 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.

In particular, we realised the following IP strategies (see examples in Table 1):

- SUMMARY of all matching restaurants with or without a User Model (UM), following (Polifroni and Walker, 2008). The approach using a UM assumes that the user has certain preferences (e.g. cheap) and only tells him about the relevant items, whereas the approach with no UM lists all the matching items.
- 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 UM).

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 will therefore concentrate on the higher level decisions (SUMMARY VS. COMPARE VS. RECOMMEND) and model these different realisations as noise in the realiser.

3.3 Supervised Baseline strategy

We analysed the WoZ data to explore the best-rated strategies (the top scoring 50%, $n = 205$) that were employed by humans for this task. Here we used a variety of Supervised Learning methods to create a model of the highly rated wizard behaviour. Please see (Rieser et al., 2009) for further details. The best performing method was Rule Induction (JRip). [2] The model achieved an accuracy of 43.19% which is significantly ($p < .001$) better than the majority baseline of always choosing SUMMARY (34.65%). [3] The resulting rule set is shown in Figure 3.

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

Figure 3: 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 input (previous action, number of database hits) that an IP module receives as input from a Dialogue Manager (DM). We further analysed the importance of different features using feature ranking and selection methods (Rieser et al., 2009), finding that the human wizards in this specific setup did not pay significant attention to any lower level features, e.g. from surface realisation, although the generated output was displayed to them (see Figure 2).

Nevertheless, note that the supervised model achieves up to 87.6% of the possible reward on this task, as we show in Section 5.2, and so can be considered a serious baseline against which to measure performance. Below, we will show that Reinforcement Learning (RL) 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.

4 The Simulation / Learning Environment

Here we "bootstrap" a simulated training environment from the WoZ data, following (Rieser and Lemon, 2008).

4.1 User Simulations

User Simulations are commonly used to train strategies for Dialogue Management, see for example (Young et al., 2007). A user simulation for NLG is very similar, in that it is a predictive model of the most likely next user act. [4] 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,

[2] The WEKA implementation of (Cohen, 1995)'s RIPPER.

[3] Note that the low accuracy is due to data sparsity and diverse behaviour of the wizards. However, in (Rieser et al., 2009) we show that this model is significantly different from the policy learned using the worse scoring 50%.

[4] Similar to the internal user models applied in recent work on POMDP (Partially Observable Markov Decision Process) dialogue managers (Young et al., 2007; Henderson and Lemon, 2008; Gasic et al., 2008) for estimation of user act probabilities.

the NLG user simulation tells us what the user is most likely to do next, *if we were to stop generating now*.

We are most 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 build user simulations using n-gram models of system (s) and user (u) acts, as first introduced by (Eckert et al., 1997). In order to account for data sparsity, we apply different *discounting* ("smoothing") techniques including *back-off*, using the CMU Statistical Language Modelling toolkit (Clarkson and Rosenfeld, 1997). We construct a **bi-gram** model[5] for the users' reactions to the system's IP structure decisions ($P(a_{u,t}|IP_{s,t})$), and a **tri-gram** (i.e. IP structure + attribute choice) model for predicting user reactions to the system's combined IP structure and attribute selection decisions: $P(a_{u,t}|IP_{s,t}, attributes_{s,t})$.

[5]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 evaluate 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. (Cuayáhuitl et al., 2005; Jung et al., 2009; Janarthanam and Lemon, 2009). We compare the raw probabilities as observed in the data with the probabilities generated by our n-gram models using different discounting techniques for each context, see table 2. 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 WittenBell method for the tri-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 GoodTuring for the tri-gram. We use the most similar user models for system training, and the most dissimilar user models for testing NLG policies, in order to test whether the learned policies are robust and adaptive to unseen dialogue contexts.

discounting method	bi-gram US	tri-gram US
WittenBell	0.086	*0.512*
GoodTuring	0.086	**0.163**
absolute	*0.091*	0.246
linear	**0.011**	0.276

Table 2: Kullback-Leibler divergence for the different User Simulations (US)

4.2 Database matches and "Focus of attention"

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 1.1. We therefore model the user's "focus of attention" as a feature in our learning experiments. This feature reflects how the different IP strategies structure information with different numbers of attributes. We implement this shift of the user's focus analogously to discovering the user's goal in Dialogue Management: every time the predicted next user act is to add in-

formation (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, the user's focus after the SUMMARY (with UM) in Table 1 is $DBhits = 10$, since the user is only interested in cheap, Indian places.

4.3 Data-driven Reward function

The reward/evaluation function is constructed from the WoZ data, using a stepwise linear regression, following the PARADISE framework (Walker et al., 2000). This model selects the features 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 (Young et al., 2007). This reflects the fact that good IP strategies should help the user to select an item ($valueUserReaction = +100$) or provide more constraints addInfo ($valueUserReaction = \pm 0$), but the user should not do anything else ($valueUserReaction = -100$). The regression in equation 1 ($R^2 = .26$) indicates that users' ratings are influenced by higher level and lower 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]):

$$\begin{aligned} Reward = (-1.2) \times \#DBhits \\ +(.121) \times valueUserReaction \\ -(1.43) \times \#sentence \end{aligned} \quad (1)$$

Note that the worst possible reward for an 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[6], in such a way that the user ends the conversation unsuccessfully. The top possible reward is achieved in the rare cases where the system can immediately present 1 item to the user using just 2 sentences, and the user then selects that item, i.e. Reward = $-(1.20 \times 1) + (.121 \times 100) - (2 \times 1.43) = 8.06$

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

5 Reinforcement Learning experiments

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} describing how good the result of action a was when performed in state s. The aim of the MDP is to maximise long-term expected reward of its decisions, resulting in a *policy* which maps each possible state to an appropriate action in that state.

We treat IP as a hierarchical joint optimisation problem, where first one of the IP structures (1-3) is chosen and then the number of attributes is decided, as shown in Figure 4. 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.)

$$\left[\begin{array}{l} \text{ACTION:} \left[\begin{array}{l} \textit{IP:} \left\{\begin{array}{l}\text{SUMMARY}\\ \text{COMPARE}\\ \text{RECOMMEND}\end{array}\right\} \left\{\text{attr: } 1\text{-}5\right\} \end{array}\right] \\ \text{STATE:} \left[\begin{array}{l} \texttt{attributes:}\left\{\texttt{1-15}\right\} \\ \texttt{sentence:}\left\{\texttt{2-18}\right\} \\ \texttt{dbHitsFocus:}\left\{\texttt{1-100}\right\} \\ \texttt{userSelect:}\left\{\texttt{0,1}\right\} \\ \texttt{userAddInfo:}\left\{\texttt{0,1}\right\} \\ \texttt{userElse:}\left\{\texttt{0,1}\right\} \end{array}\right] \end{array}\right]$$

Figure 4: State-Action space for the RL-NLG problem

States are represented as sets of NLG dialogue context features. The state space comprises "lower-level" features about the realiser behaviour (two discrete features representing the number of attributes and sentences generated so far) and three binary features representing the user's predicted next action, as well as "high-level" features pro-

vided by the DM (e.g. current database hits in the user's focus (`dbHitsFocus`)). We trained the policy using the SHARSHA algorithm (Shapiro and Langley, 2002) with linear function approximation (Sutton and Barto, 1998), and the simulation environment described in Section 4. The policy was trained for 60,000 iterations.

5.1 Experimental Set-up

We compare the learned strategies against the *WoZ baseline* as described in Section 3.3. For attribute selection we choose a majority baseline (randomly choosing between 3 or 4 attributes) since the attribute selection models learned by Supervised Learning on the WoZ data didn't show significant improvements.

For training, we used the user simulation model most similar to the data, see Section 4.1. For testing, we test with the *different* user simulation model (the one which is most dissimilar to the data).

We first investigate how well IP structure (without attribute choice) can be learned in increasingly complex generation **scenarios**. A generation scenario is a combination of a particular kind of NLG realiser (template vs. stochastic) along with different levels of variation introduced by certain features of the dialogue context. In general, the stochastic realiser introduces more variation in lower level features than the template-based realiser. The Focus model introduces more variation with respect to #DBhits and #attributes as described in Section 4.2. We therefore investigate the following cases:

1.1. IP structure choice, 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, 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 3.2.

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

2.1. IPS+Attr choice, Template realiser: Predicted next user action varies according to tri-gram ($P(a_{u,t}|IP_{s,t}, attributes_{s,t})$) model; Number of sentences per IP structure set to default.

2.2. IPS+Attr choice, Template realiser+Focus model: Tri-gram user simulation with Template realiser and Focus of attention model with respect to #DBhits and #attributes as described in section 4.2.

2.3. IPS+Attr choice, Stochastic realiser: Tri-gram user simulation with sentence/attribute relationship according to Stochastic realiser as described in Section 3.2.

2.4. IPS+Attr choice, Stochastic realiser+Focus: i.e. the full model = Predicted next user action varies according to tri-gram model+ Focus of attention model + Sentence/attribute relationship according to stochastic realiser.

5.2 Results

We compare the average final reward (see Equation 1) 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 3. 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 gained by the individual policies, and the raw percentage improvement of the RL policy. Note that the best possible (100%) reward can only be gained in rare cases (see Section 4.3).

The learned RL policies show that lower level features are important in gaining significant improvements over the baseline. The more complex the scenario, the harder it is to gain higher rewards for the policies in general (as more variation is introduced), but the relative improvement in rewards also increases with complexity: the baseline does not adapt well to the variations in lower level features whereas RL learns to adapt to the more challenging scenarios. [7]

An overview of the range of different IP strategies learned for each setup can be found in Table 4. Note that these strategies are context-dependent: the learner chooses how to proceed dependent on

[7] Note, that the baseline does reasonably well in scenarios with variation introduced by only higher level features (e.g. scenario 2.2).

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%

Table 3: Test results for 1000 dialogues, where *** denotes that the RL policy is significantly ($p < .001$) better than the Baseline policy.

the features in the state space at each generation step.

Scenario	strategies learned
1.1	RECOMMEND COMPARE COMPARE+RECOMMEND SUMMARY SUMMARY+COMPARE SUMMARY+RECOMMEND SUMMARY+COMPARE+RECOMMEND.
1.2	RECOMMEND COMPARE COMPARE+RECOMMEND SUMMARY SUMMARY+COMPARE SUMMARY+RECOMMEND SUMMARY+COMPARE+RECOMMEND.
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)

Table 4: RL strategies learned for the different scenarios, where (n) denotes the number of attributes generated.

For example, 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. Also see Table 4. Note that the WoZ strategy behaves as described in Figure 3.

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), because the overall generated utterance (final #sentences) is still relatively short. Otherwise the policy is similar to the one for scenario 1.1.

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 database hits are 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 (tri-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 4.2. 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 results 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) num-

ber 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 3). 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, 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).

6 Conclusion

We have presented a new data-driven method for Information Presentation (IP) in Spoken Dialogue Systems using a statistical optimisation framework for content structure planning and attribute selection. This work is the first to apply a data-driven optimisation method to the IP decision space, and to show the utility of both lower-level and higher-level features for this problem.

We collected data in a Wizard-of-Oz (WoZ) experiment and showed that human "wizards" mostly pay attention to 'high-level' features from Dialogue Management. The WoZ data was used to build statistical models of user reactions to IP strategies, and a data-driven reward function for Reinforcement Learning (RL). We show that lower level features significantly influence users' ratings of IP strategies. We compared a model of human behaviour (the 'human wizard baseline') against policies optimised using Reinforcement Learning, in a variety of scenarios. Our optimised policies significantly outperform the IP structuring and attribute selection present in the WoZ data, especially when performing in complex generation scenarios which require adaptation to, e.g. number of database results, utterance length, etc. While the human wizards were able to attain up to 87.6% of the possible reward on this task, the RL policies are significantly better in 5 out of 6 scenarios, gaining up to 91.5% of the total possible reward.

We have also shown that adding predictive "lower level" features, e.g. from the NLG realiser and a user reaction model, is important for learning optimal IP strategies according to user preferences. Future work could include the predicted TTS quality (Boidin et al., 2009) as a feature.

We are now working on testing the learned policies with real users, outside of laboratory conditions, using a restaurant-guide SDS, deployed as a VOIP service. Previous work in SDS has shown that results for Dialogue Management obtained with simulated users are able to transfer to evaluations with real users (Lemon et al., 2006).

This methodology 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. The data-driven planning method applied here promises a significant upgrade in the performance of generation modules, and thereby of Spoken Dialogue Systems in general.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216594 (CLASSiC project `www.classic-project.org`) and from the EPSRC, project no. EP/G069840/1.

References

Cedric Boidin, Verena Rieser, Lonneke van der Plas, Oliver Lemon, and Jonathan Chevelu. 2009. Predicting how it sounds: Re-ranking alternative inputs to TTS using latent variables (forthcoming). In *Proc. of Interspeech/ICSLP, Special Session on Machine Learning for Adaptivity in Spoken Dialogue Systems*.

Grace Chung. 2004. Developing a flexible spoken dialog system using simulation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

P.R. Clarkson and R. Rosenfeld. 1997. Statistical Language Modeling Using the CMU-Cambridge Toolkit. In *Proc. of ESCA Eurospeech*.

William W. Cohen. 1995. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning (ICML)*.

Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon, and Hiroshi Shimodaira. 2005. Human-computer dialogue simulation using hidden markov models. In *Proc. of the IEEE workshop on Automatic Speech Recognition and Understanding (ASRU)*.

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

W. Eckert, E. Levin, and R. Pieraccini. 1997. User modeling for spoken dialogue system evaluation. In *Proc. of the IEEE workshop on Automatic Speech Recognition and Understanding (ASRU)*.

M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and S. Young. 2008. Training and Evaluation of the HIS POMDP Dialogue System in Noise. In *Proc. of SIGdial Workshop on Discourse and Dialogue*.

James Henderson and Oliver Lemon. 2008. Mixture Model POMDPs for Efficient Handling of Uncertainty in Dialogue Management. In *Proc. of ACL*.

Srinivasan Janarthanam and Oliver Lemon. 2009. A Two-tier User Simulation Model for Reinforcement Learning of Adaptive Referring Expression Generation Policies. In *Proc. of SIGdial*.

Srini Janarthanam and Oliver Lemon. 2010. Learning to adapt to unknown users: Referring expression generation in spoken dialogue systems. In *Proceedings of ACL*.

Sangkeun Jung, Cheongjae Lee, Kyungduk Kim, Minwoo Jeong, and Gary Geunbae Lee. 2009. Data-driven user simulation for automated evaluation of spoken dialog systems. *Computer, Speech & Language*, 23:479–509.

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

Oliver Lemon, Kallirroi Georgila, and James Henderson. 2006. Evaluating Effectiveness and Portability of Reinforcement Learned Dialogue Strategies with real users: the TALK TownInfo Evaluation. In *IEEE/ACL Spoken Language Technology*.

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

Oliver Lemon. 2010. Learning what to say and how to say it: joint optimization of spoken dialogue management and Natural Language Generation. *Computer, Speech & Language*, to appear.

Xingkun Liu, Verena Rieser, and Oliver Lemon. 2009. 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*.

Crystal Nakatsu. 2008. Learning contrastive connectives in sentence realization ranking. In *Proc. of SIGdial Workshop on Discourse and Dialogue*.

Joseph Polifroni and Marilyn Walker. 2006. Learning database content for spoken dialogue system design. In *Proc. of the IEEE/ACL workshop on Spoken Language Technology (SLT)*.

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

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

Verena Rieser and Oliver Lemon. 2009. Natural Language Generation as Planning Under Uncertainty for Spoken Dialogue Systems. In *Proc. of EACL*.

Verena Rieser, Xingkun Liu, and Oliver Lemon. 2009. Optimal Wizard NLG Behaviours in Context. Technical report, Deliverable 4.2, CLASSiC Project.

Dan Shapiro and P. Langley. 2002. Separating skills from preference: Using learning to program by reward. In *Proc. of the 19th International Conference on Machine Learning (ICML)*.

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

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

Kees van Deemter. 2009. What game theory can do for NLG: the case of vague language. In *12th European Workshop on Natural Language Generation (ENLG)*.

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

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

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

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

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

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