Generation under Uncertainty

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

Srini Janarthanam
Edinburgh University
Edinburgh, United Kingdom
s.janarthanam@ed.ac.uk

Verena Rieser
Edinburgh University
Edinburgh, United Kingdom
vrieser@inf.ed.ac.uk

Abstract

We invite the research community to consider challenges for NLG which arise from uncertainty. NLG systems should be able to adapt to their audience and the generation environment in general, but often the important features for adaptation are not known precisely. We explore generation challenges which could employ simulated environments to study NLG which is adaptive under uncertainty, and suggest possible metrics for such tasks. It would be particularly interesting to explore how different planning approaches to NLG perform in challenges involving uncertainty in the generation environment.

1 Introduction

We would like to highlight the design of NLG systems for environments where there may be incomplete or faulty information, where actions may not always have the same results, and where there may be tradeoffs between the different possible outcomes of actions and plans.

There are various sources of uncertainty in systems which employ NLG techniques, for example:

- the current state of the user / audience (e.g. their knowledge, preferred vocabulary, goals, preferences....),
- the likely user reaction to the generated output,
- the behaviour of related components (e.g. a surface realiser, or TTS module),
- noise in the environment (for spoken output),
- ambiguity of the generated output.

The problem here is to generate output that takes these types of uncertainty into account appropriately. For example, you may need to choose a referring expression for a user, even though you are not sure whether they are an expert or novice in the domain. In addition, the next time you speak to that user, you need to adapt to new information you have gained about them (Janarthanam and Lemon, 2010). The issue of uncertainty for referring expression generation has been discussed before by (Reiter, 1991; Horacek, 2005).

Another example is in planning an Information Presentation for a user, when you cannot know with certainty how they will respond to it (Rieser and Lemon, 2009; Rieser et al., 2010). In the worst case, you may even be uncertain about the user's goals or information needs (as in "POMDP" approaches to dialogue management (Young et al., 2009; Henderson and Lemon, 2008a)), but you still need to generate output for them in an appropriate way.

In particular, in interactive applications of NLG:

- each NLG action *changes* the environment state or context,
- the effect of each NLG action is *uncertain*.

Several recent approaches describe NLG tasks as different kinds of planning, e.g. (Koller and Petrick, 2008; Rieser et al., 2010; Janarthanam and Lemon, 2010), or as contextual decision making according to a cost function (van Deemter, 2009). It would be very interesting to explore how different approaches perform in NLG problems where different types of uncertainty are present in the generation environment.

In the following we discuss possible generation challenges arising from such considerations, which we hope will lead to work on an agreed shared challenge in this research community. In section 2 we briefly review recent work showing

that simulated environments can be used to evaluate generation under uncertainty, and in section 3 we discuss some possible metrics for such tasks. Section 4 concludes by considering how a useful generation challenge could be constructed using similar methods.

2 Generation in Uncertain Simulated Environments

Finding the best (or "optimal") way to generate under uncertainty requires exploring the possible outcomes of actions in stochastic environments. Therefore, related research on Dialogue Strategy learning has used data-driven simulated environments as a cheap and efficient way to explore uncertainty (Lemon and Pietquin, 2007). However, building good simulated environments is a challenge in its own right, as we illustrate in the following using the examples of Information Presentation and Referring Expression Generation. We also point out the additional challenges these simulations have to face when being used for NLG.

2.1 User Simulations for Information Presentation

User Simulations can provide a model of probable, but uncertain, user reactions to NLG actions, and we propose that they are a useful potential direction for exploring and evaluate different approaches to handling uncertainty in generation.

User Simulations are commonly used to train strategies for Dialogue Management, see for example (Young et al., 2007). A user simulation for Information Presentation is very similar, in that it is a predictive model of the most likely next user act. [1] 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, this NLG user simulation tells us what the user is most likely to do next, *if we were to stop generating now*.

In addition to the challenges of building user simulations for learning Dialogue policies, e.g. modelling, evaluation, and available data sets (Lemon and Pietquin, 2007), a crucial decision for NLG is the level of detail needed to train sensible policies. While high-level dialogue act descriptions may be sufficient for dialogue policies, NLG decisions may require a much finer level of detail. The finer the required detail of user reactions, the more data is needed to build data-driven simulations.

For content selection in Information Presentation tasks (choosing presentation strategy and number of attributes), for example, the level of description can still be fairly abstract. We were most interested in probability distributions over the following possible 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 have built 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 automatic *back-off*, using the CMU Statistical Language Modelling toolkit (Clarkson and Rosenfeld, 1997). For example we have constructed a **bi-**

[1] 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, 2008b; Gasic et al., 2008) for estimation of user act probabilities.

gram model[2] 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: $P(a_{u,t}|IP_{s,t}, attributes_{s,t})$.

We have 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. (Cuayáhuitl et al., 2005; Jung et al., 2009; Janarthanam and Lemon, 2009). We compared the raw probabilities as observed in the data with the probabilities generated by our n-gram models using different discounting techniques for each context. 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 (Rieser et al., 2010).

2.2 Other Simulated Components

In some systems, NLG decisions may also depend on related components, such as the database, subsequent generation steps, or the Text-to-Speech module for spoken generation. Building simulations for these components to capture their inherent uncertainty, again, is an interesting challenge.

For example, one might want to adapt the generated output according to the predicted TTS quality. Therefore, one needs a model of the expected/ predicted TTS quality for a TTS engine (Boidin et al., 2009).

Furthermore, NLG decisions might be inputs to a stochastic sentence realiser, such as SPaRKy (Stent et al., 2004). However, one might not have a fully trained stochastic sentence realiser for this domain (yet). In (Rieser et al., 2010) we therefore modelled the variance as observed in the top ranking SPaRKy examples.

2.3 Generating Referring Expressions under uncertainty

In this section, we present an example user simulation (US) model, that simulates the dialogue behaviour of users who react to referring expressions depending on their domain knowledge. These external simulation models are different from internal user models used by dialogue systems. In particular, such models must be sensitive to a system's choices of referring expressions. The simulation has a statistical distribution of in-built knowledge profiles that determines the dialogue behaviour of the user being simulated. Uncertainty arises because if the user does not know a referring expression, then he is more *likely* to request clarification. If the user is able to interpret the referring expressions and identify the references then he is more likely to follow the system's instruction. This behaviour is simulated by the action selection models described below.

The user simulation (US) receives the system action $A_{s,t}$ and its referring expression choices $REC_{s,t}$ at each turn. The US responds with a user action $A_{u,t}$ (u denoting user). This can either be a clarification request (cr) or an instruction response (ir). We used two kinds of action selection models: a corpus-driven statistical model and a hand-coded rule-based model.

2.4 Corpus-driven action selection model

The user simulation (US) receives the system action $A_{s,t}$ and its referring expression choices $REC_{s,t}$ at each turn. The US responds with a user action $A_{u,t}$ (u denoting user). This can either be a clarification request (cr) or an instruction response (ir). The US produces a clarification request cr based on the class of the referent $C(R_i)$, type of the referring expression T_i, and the current domain knowledge of the user for the referring expression $DK_{u,t}(R_i, T_i)$. Domain entities whose jargon expressions raised clarification requests in the corpus were listed and those that had more than the mean number of clarification requests were classified as `difficult` and others as `easy` entities (for example, "power adaptor" is `easy` - all users understood this expression, "broadband filter" is `difficult`). Clarification requests are produced using the following model.

$$P(A_{u,t} = cr(R_i, T_i)|C(R_i), T_i, DK_{u,t}(R_i, T_i))$$
$$\text{where } (R_i, T_i) \in REC_{s,t}$$

One should note that the actual literal expression is not used in the transaction. Only the entity that it is referring to (R_i) and its type (T_i) are used. However, the above model simulates the process of interpreting and resolving the expression and identifying the domain entity of interest in the instruction. The user identification of the entity is signified when there is no clarification request produced (i.e. $A_{u,t} = none$). When no clarification

[2]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 set of attributes selected by the system at t.

request is produced, the environment action $EA_{u,t}$ is generated using the following model.

$$P(EA_{u,t}|A_{s,t}) \text{ if } A_{u,t}! = cr(R_i, T_i)$$

Finally, the user action is an instruction response which is determined by the system action $A_{s,t}$. Instruction responses can be either $provide\_info$, $acknowledgement$ or $other$ based on the system's instruction.

$$P(A_{u,t} = ir|EA_{u,t}, A_{s,t})$$

All the above models were trained on our corpus data using *maximum likelihood estimation* and smoothed using a variant of *Witten-Bell discounting*. According to the data, clarification requests are much more likely when jargon expressions are used to refer to the referents that belong to the `difficult` class and which the user doesn't know about. When the system uses expressions that the user knows, the user generally responds to the instruction given by the system. These user simulation models have been evaluated and found to produce behaviour that is very similar to the original corpus data, using the Kullback-Leibler divergence metric (Janarthanam and Lemon, 2010).

3 Metrics

Here we discuss some possible evaluation metrics that will allow different approaches to NLG under uncertainty to be compared. We envisage that other metrics should be explored, in particular those measuring adaptivity of various types.

3.1 Adaptive Information Presentation

Given a suitable corpus, a data-driven evaluation function can be constructed, using a stepwise linear regression, following the PARADISE framework (Walker et al., 2000).

For example, in (Rieser et al., 2010) we build a model which selects the features which significantly influenced the users' ratings for NLG strategies in a Wizard-of-Oz study. 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 Information Presentation 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 $\#sentence$ was in the range [2-18]):

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

3.2 Measuring Adaptivity of Referring Expressions

We have also designed a metric for the goal of adapting referring expressions to each user's domain knowledge. We present the Adaptation Accuracy score AA that calculates how accurately the agent chose the expressions for each referent r, with respect to the user's knowledge. Appropriateness of an expression is based on the user's knowledge of the expression. So, when the user knows the jargon expression for r, the appropriate expression to use is jargon, and if s/he doesn't know the jargon, an descriptive expression is appropriate. Although the user's domain knowledge is dynamically changing due to learning, we base appropriateness on the initial state, because our objective is to adapt to the initial state of the user $DK_{u,initial}$. However, in reality, designers might want their system to account for user's changing knowledge as well. We calculate accuracy per referent RA_r as the ratio of number of appropriate expressions to the total number of instances of the referent in the dialogue. We then calculate the overall mean accuracy over all referents as shown below.

$$RA_r = \frac{\#(appropriate\_expressions(r))}{\#(instances(r))}$$
$$AdaptationAccuracy AA = \frac{1}{\#(r)} \Sigma_r RA_r$$

4 Conclusion

We have invited the research community to consider challenges for NLG which arise from uncertainty. We argue that NLG systems, like dialogue managers, should be able to adapt to their audience and the generation environment. However, often the important features for adaptation are not precisely known. We then summarised 2 potential

directions for such challenges – example generation tasks which employ simulated uncertain environments to study adaptive NLG, and discussed some possible metrics for such tasks. We hope that this will lead to discussions on a shared challenge allowing comparison of different approaches to NLG with respect to how well they handle uncertainty.

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*.

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

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)*.

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. 2008a. Mixture Model POMDPs for Efficient Handling of Uncertainty in Dialogue Management. In *Proceedings of ACL*.

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

Helmut Horacek. 2005. Generating referential descriptions under conditions of uncertainty. In *ENLG*.

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*. (to appear).

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 and Olivier Pietquin. 2007. Machine learning for spoken dialogue systems. In *Interspeech*.

E. Reiter. 1991. Generating Descriptions that Exploit a User's Domain Knowledge. In R. Dale, C. Mellish, and M. Zock, editors, *Current Research in Natural Language Generation*, pages 257–285. Academic Press.

Verena Rieser and Oliver Lemon. 2009. Natural language generation as planning under uncertainty for spoken dialogue systems. In *EACL*.

Verena Rieser, Oliver Lemon, and Xingkun Liu. 2010. Optimising information presentation for spoken dialogue systems. In *Proceedings of ACL*. (to appear).

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

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).

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

S. Young, M. Gašić, S. Keizer, F. Mairesse, B. Thomson, and K. Yu. 2009. The Hidden Information State model: a practical framework for POMDP based spoken dialogue management. *Computer Speech and Language*. To appear.