Hierarchical Reinforcement Learning and Hidden Markov Models for Task-Oriented Natural Language Generation

Nina Dethlefs
Department of Linguistics,
University of Bremen
dethlefs@uni-bremen.de

Heriberto Cuayáhuitl
German Research Centre for Artificial Intelligence
(DFKI), Saarbrücken
heriberto.cuayahuitl@dfki.de

Abstract

Surface realisation decisions in language generation can be sensitive to a language model, but also to decisions of content selection. We therefore propose the joint optimisation of content selection and surface realisation using Hierarchical Reinforcement Learning (HRL). To this end, we suggest a novel reward function that is induced from human data and is especially suited for surface realisation. It is based on a generation space in the form of a Hidden Markov Model (HMM). Results in terms of task success and human-likeness suggest that our unified approach performs better than greedy or random baselines.

1 Introduction

Surface realisation decisions in a Natural Language Generation (NLG) system are often made according to a language model of the domain (Langkilde and Knight, 1998; Bangalore and Rambow, 2000; Oh and Rudnicky, 2000; White, 2004; Belz, 2008). However, there are other linguistic phenomena, such as alignment (Pickering and Garrod, 2004), consistency (Halliday and Hasan, 1976), and variation, which influence people's assessment of discourse (Levelt and Kelter, 1982) and generated output (Belz and Reiter, 2006; Foster and Oberlander, 2006). Also, in dialogue the most likely surface form may not always be appropriate, because it does not correspond to the user's information need, the user is confused, or the most likely sequence is infelicitous with respect to the dialogue history. In such cases, it is important to optimise surface realisation in a unified fashion with content selection. We suggest to use Hierarchical Reinforcement Learning (HRL) to achieve this. Reinforcement Learning (RL) is an attractive framework for optimising a sequence of decisions given incomplete knowledge of the environment or best strategy to follow (Rieser et al., 2010; Janarthanam and Lemon, 2010). HRL has the additional advantage of scaling to large and complex problems (Dethlefs and Cuayáhuitl, 2010). Since an HRL agent will ultimately learn the behaviour it is rewarded for, the reward function is arguably the agent's most crucial component. Previous work has therefore suggested to learn a reward function from human data as in the PARADISE framework (Walker et al., 1997). Since PARADISE-based reward functions typically rely on objective metrics, they are not ideally suited for surface realisation, which is more dependend on linguistic phenomena, e.g. frequency, consistency, and variation. However, linguistic and psychological studies (cited above) show that such phenomena are indeed worth modelling in an NLG system. The contribution of this paper is therefore to induce a reward function from human data, specifically suited for surface generation. To this end, we train HMMs (Rabiner, 1989) on a corpus of grammatical word sequences and use them to inform the agent's learning process. In addition, we suggest to optimise surface realisation and content selection decisions in a joint, rather than isolated, fashion. Results show that our combined approach generates more successful and human-like utterances than a greedy or random baseline. This is related to Angeli et al. (2010), who also address interdependent decision making, but do not use an optmisation framework. Since language models in our approach can be obtained for any domain for which corpus data is available, it generalises to new domains with limited effort and reduced development

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 654–659,
Portland, Oregon, June 19-24, 2011.

Utterance
string="turn around and go out", time='20:54:55'
Utterance_type
content='orientation,destination' [straight, path, direction]
navigation_level='low' [high]
User
user_reaction='perform_desired_action'
[perform_undesired_action, wait, request_help]
user_position='on_track' [off_track]

Figure 1: Example annotation: alternative values for attributes are given in square brackets.

time. For related work on using graphical models for language generation, see e.g., Barzilay and Lee (2002), who use lattices, or Mairesse et al. (2010), who use dynamic Bayesian networks.

2 Generation Spaces

We are concerned with the generation of navigation instructions in a virtual 3D world as in the GIVE scenario (Koller et al., 2010). In this task, two people engage in a 'treasure hunt', where one participant navigates the other through the world, pressing a sequence of buttons and completing the task by obtaining a trophy. The GIVE-2 corpus (Gargett et al., 2010) provides transcripts of such dialogues in English and German. For this paper, we complemented the English dialogues of the corpus with a set of semantic annotations,[1] an example of which is given in Figure 1. This example also exemplifies the type of utterances we generate. The input to the system consists of semantic variables comparable to the annotated values, the output corresponds to strings of words. We use HRL to optimise decisions of content selection ('what to say') and HMMs for decisions of surface realisation ('how to say it'). **Content selection** involves whether to use a low-, or high-level navigation strategy. The former consists of a sequence of primitive instructions ('go straight', 'turn left'), the latter represents contractions of sequences of low-level instructions ('head to the next room'). Content selection also involves choosing a level of detail for the instruction corresponding to the user's information need. We evaluate the learnt content selection decisions in terms of task success. For **surface realisation**, we use HMMs to inform the HRL agent's learning process. Here we address the one-to-many relationship arising between a semantic form (from the content selection stage) and its possible realisations. Semantic forms of instructions have an average of 650 surface realisations, including syntactic and lexical variation, and decisions of granularity. We refer to the set of alternative realisations of a semantic form as its 'generation space'. In surface realisation, we aim to optimise the tradeoff between alignment and consistency (Pickering and Garrod, 2004; Halliday and Hasan, 1976) on the one hand, and variation (to improve text quality and readability) on the other hand (Belz and Reiter, 2006; Foster and Oberlander, 2006) in a $50/50$ distribution. We evaluate the learnt surface realisation decisions in terms of similarity with human data.

Note that while we treat content selection and surface realisation as separate NLG tasks, their optimisation is achieved jointly. This is due to a tradeoff arising between the two tasks. For example, while surface realisation decisions that are optimised solely with respect to a language model tend to favour frequent and short sequences, these can be inappropriate according to the user's information need (because they are unfamiliar with the navigation task, or are confused or lost). In such situations, it is important to treat content selection and surface realisation as a unified whole. Decisions of both tasks are inextricably linked and we will show in Section 5.2 that their joint optimisation leads to better results than an isolated optimisation as in, for example, a two-stage model.

3 NLG Using HRL and HMMs

3.1 Hierarchical Reinforcement Learning

The idea of *language generation as an optimisation problem* is as follows: given a set of generation states, a set of actions, and an objective reward function, an optimal generation strategy maximises the objective function by choosing the actions leading to the highest reward for every reached state. Such states describe the system's knowledge about the generation task (e.g. content selection, navigation strategy, surface realisation). The action set describes the system's capabilities (e.g. *'use high level navigation strategy'*, *'use imperative mood'*, etc.). The reward function assigns a numeric value for each action taken. In this way, language gen-

[1] The annotations are available on request.

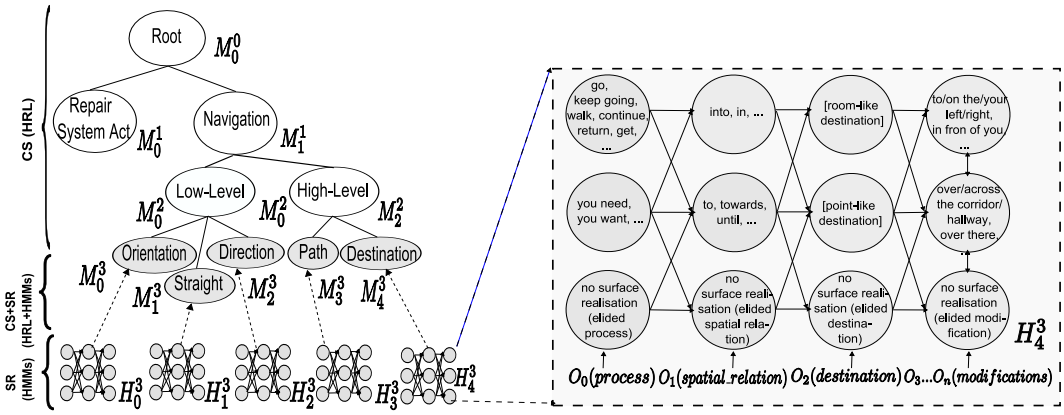

Figure 2: Hierarchy of learning agents (left), where shaded agents use an HMM-based reward function. The top three layers are responsible for content selection (CS) decisions and use HRL. The shaded agents in the bottom use HRL with an HMM-based reward function and joint optimisation of content selection and surface realisation (SR). They provide the observation sequence to the HMMs. The HMMs represent generation spaces for surface realisation. An example HMM, representing the generation space of 'destination' instructions, is shown on the right.

eration can be seen as a finite sequence of states, actions and rewards $\{s_0, a_0, r_1, s_1, a_1, ..., r_{t-1}, s_t\}$, where the goal is to find an optimal strategy automatically. To do this we use RL with a divide-and-conquer approach to optimise a hierarchy of generation policies rather than a single policy. The hierarchy of RL agents consists of L levels and N models per level, denoted as M_j^i, where $j \in \{0, ..., N-1\}$ and $i \in \{0, ..., L-1\}$. Each agent of the hierarchy is defined as a Semi-Markov Decision Process (SMDP) consisting of a 4-tuple $< S_j^i, A_j^i, T_j^i, R_j^i >$. S_j^i is a set of states, A_j^i is a set of actions, T_j^i is a transition function that determines the next state s' from the current state s and the performed action a, and R_j^i is a reward function that specifies the reward that an agent receives for taking an action a in state s lasting τ time steps. The random variable τ represents the number of time steps the agent takes to complete a subtask. Actions can be either primitive or composite. The former yield single rewards, the latter correspond to SMDPs and yield cumulative discounted rewards. The goal of each SMDP is to find an optimal policy that maximises the reward for each visited state, according to ${\pi^*}_j^i(s) = \arg\max_{a \in A_j^i} {Q^*}_j^i(s,a)$, where $Q_j^{*i}(s,a)$ specifies the expected cumulative reward for executing action a in state s and then following policy ${\pi^*}_j^i$. We use HSMQ-Learning (Dietterich, 1999) to learn a hierarchy of generation policies.

3.2 Hidden Markov Models for NLG

The idea of representing the generation space of a surface realiser as an HMM can be roughly defined as the converse of POS tagging, where an input string of words is mapped onto a hidden sequence of POS tags. Our scenario is as follows: given a set of (specialised) semantic symbols (e.g., 'actor', 'process', 'destination'),[2] what is the most likely sequence of words corresponding to the symbols? Figure 2 provides a graphic illustration of this idea. We treat states as representing words, and sequences of states $i_0 ... i_n$ as representing phrases or sentences. An observation sequence $o_0 ... o_n$ consists of a finite set of semantic symbols specific to the instruction type (i.e., 'destination', 'direction', 'orientation', 'path', 'straight'). Each symbol has an observation likelihood $b_i(o_t)$, which gives the probability of observing o in state i at time t. The transition and emission probabilities are learnt during training using the Baum-Welch algorithm. To design an HMM from the corpus data, we used the ABL algorithm (van Zaanen, 2000), which aligns strings based on Minimum Edit Distance, and induces a context-free grammar from the aligned examples. Subsequently, we constructed the HMMs from the CFGs, one for each instruction type, and trained them on the annotated data.

[2] Utterances typically contain five to ten semantic categories.

3.3 An HMM-based Reward Function Induced from Human Data

Due to its unique function in an RL framework, we suggest to induce a reward function for surface realisation from human data. To this end, we create and train HMMs to represent the generation space of a particular surface realisation task. We then use the forward probability, derived from the Forward algorithm, of an observation sequence to inform the agent's learning process.

$$
\mathrm{r} = \begin{cases}
0 & \text{for} & \text{reaching the goal state} \\
+1 & \text{for} & \text{a desired semantic choice or maintaining an equal distribution of alignment and variation} \\
-2 & \text{for} & \text{executing action } a \text{ and remaining in the same state } s = s' \\
P(w_0 ... w_n) & \text{for} & \text{for reaching a goal state corresponding to word sequence } w_0 ... w_n \\
-1 & & \text{otherwise.}
\end{cases}
$$

Whenever the agent has generated a word sequence $w_0 ... w_n$, the HMM assigns a reward corresponding to the likelihood of observing the sequence in the data. In addition, the agent is rewarded for short interactions at maximal task success[3] and optimal content selection (cf. Section 2). Note that while reward $P(w_0 ... w_n)$ applies only to surface realisation agents $M^3_{0...4}$, the other rewards apply to all agents of the hierarchy.

4 Experimental Setting

We test our approach using the (hand-crafted) hierarchy of generation subtasks in Figure 2. It consists of a root agent (M^0_0), and subtasks for low-level (M^2_0) and high-level (M^2_1) navigation strategies (M^1_1), and for instruction types 'orientation' (M^3_0), 'straight' (M^3_1), 'direction' (M^3_2), 'path' (M^3_3) and destination' (M^3_4). Models $M^3_{0...4}$ are responsible for surface generation. They will be trained using HRL with an HMM-based reward function induced from human data. All other agents use hand-crafted rewards. Finally, subtask M^1_0 can repair a previous system utterance. The states of the agent contain all situational and linguistic information relevant to its decision making, e.g., the spatial environment, discourse history, and status of grounding.[4] Due to space constraints, please see Dethlefs et al. (2011) for the full state-action space. We distinguish primitive actions (corresponding to single generation decisions) and composite actions (corresponding to generation subtasks (Fig. 2)).

[3]Task success is addressed by that each utterance needs to be 'accepted' by the user (cf. Section 5.1).

5 Experiments and Results

5.1 The Simulated Environment

The simulated environment contains two kinds of uncertainties: (1) uncertainty regarding the state of the environment, and (2) uncertainty concerning the user's reaction to a system utterance. The first aspect is represented by a set of contextual variables describing the environment, [5] and user behaviour.[6] Altogether, this leads to 115 thousand different contextual configurations, which are estimated from data (cf. Section 2). The uncertainty regarding the user's reaction to an utterance is represented by a Naive Bayes classifier, which is passed a set of contextual features describing the situation, mapped with a set of semantic features describing the utterance.[7] From these data, the classifier specifies the most likely user reaction (after each system act) of *perform_desired_action, perform_undesired_action, wait* and *request_help*.[8] The classifier was trained on the annotated data and reached an accuracy of 82% in a cross-corpus validation using a 60%-40% split.

5.2 Comparison of Generation Policies

We trained three different generation policies. The **learnt policy** optimises content selection and surface realisation decisions in a unified fashion, and is informed by an HMM-based generation space reward function. The **greedy policy** is informed only by the HMM and always chooses the most

[4]An example for the state variables of model M^1_1 are the annotation values in Fig. 1 which are used as the agent's knowledge base. Actions are 'choose easy route', 'choose short route', 'choose low level strategy', 'choose high level strategy'.

[5]previous system act, route length, route status (known/unknown), objects within vision, objects within dialogue history, number of instructions, alignment(proportion)

[6]previous user reaction, user position, user waiting(true/false), user type(explorative/hesitant/medium)

[7]navigation level(high / low), abstractness(implicit / explicit), repair(yes / no), instruction type(destination / direction / orientation / path / straight)

[8]User reactions measure the system's task success.

likely sequence independent of content selection. The **valid sequence policy** generates any grammatical sequence. All policies were trained for 20000 episodes.[9] Figure 3, which plots the average rewards of all three policies (averaged over ten runs), shows that the 'learnt' policy performs best in terms of task success by reaching the highest overall rewards over time. An absolute comparison of the average rewards (rescaled from 0 to 1) of the last 1000 training episodes of each policy shows that greedy improves 'any valid sequence' by 71%, and learnt improves greedy by 29% (these differences are significant at $p < 0.01$). This is due to the learnt policy showing more adaptation to contextual features than the greedy or 'valid sequence' policies. To evaluate human-likeness, we compare instructions (i.e. word sequences) using Precision-Recall based on the F-Measure score, and dialogue similarity based on the Kulback-Leibler (KL) divergence (Cuayáhuitl et al., 2005). The former shows how the texts generated by each of our generation policies compare to human-authored texts in terms of precision and recall. The latter shows how similar they are to human-authored texts. Table 1 shows results of the comparison of two human data sets 'Real1' vs 'Real2' and both of them together against our different policies. While the greedy policy receives higher F-Measure scores, the learnt policy is most similar to the human data. This is due to variation: in contrast to greedy behaviour, which always exploits the most likely variant, the learnt policy varies surface forms. This leads to lower F-Measure scores, but achieves higher similarity with human authors. This ultimately is a desirable property, since it enhances the quality and naturalness of our instructions.

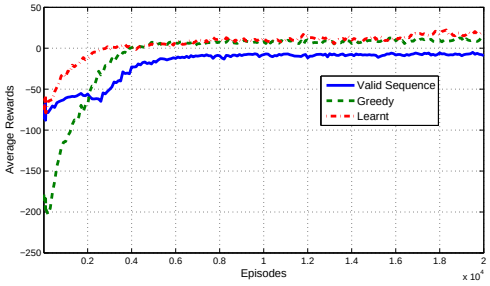

Figure 3: Performance of 'learnt', 'greedy', and 'any valid sequence' generation behaviours (average rewards).

Compared Policies	F-Measure	KL-Divergence
Real1 - Real2	0.58	1.77
Real - 'Learnt'	0.40	2.80
Real - 'Greedy'	0.49	4.34
Real - 'Valid Seq.'	0.0	10.06

Table 1: Evaluation of generation behaviours with Precision-Recall and KL-divergence.

6 Conclusion

We have presented a novel approach to optimising surface realisation using HRL. We suggested to inform an HRL agent's learning process by an HMM-based reward function, which was induced from data and in which the HMM represents the generation space of a surface realiser. We also proposed to jointly optimise surface realisation and content selection to balance the tradeoffs of (a) frequency in terms of a language model, (b) alignment/consistency vs variation, (c) properties of the user and environment. Results showed that our hybrid approach outperforms two baselines in terms of task success and human-likeness: a greedy baseline acting independent of content selection, and a random 'valid sequence' baseline. Future work can transfer our approach to different domains to confirm its benefits. Also, a detailed human evaluation study is needed to assess the effects of different surface form variants. Finally, other graphical models besides HMMs, such as Bayesian Networks, can be explored for informing the surface realisation process of a generation system.

Acknowledgments

Thanks to the German Research Foundation DFG and the Transregional Collaborative Research Centre SFB/TR8 'Spatial Cognition' and the EU-FP7 project ALIZ-E (ICT-248116) for partial support of this work.

[9]For training, the step-size parameter α (one for each SMDP) , which indicates the learning rate, was initiated with 1 and then reduced over time by $\alpha = \frac{1}{1+t}$, where t is the time step. The discount rate γ, which indicates the relevance of future rewards in relation to immediate rewards, was set to 0.99, and the probability of a random action ϵ was 0.01. See Sutton and Barto (1998) for details on these parameters.

References

Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 502–512.

Srinivas Bangalore and Owen Rambow. 2000. Exploiting a probabilistic hierarchical model for generation. In *Proceedings of the 18th Conference on Computational Linguistics (ACL) - Volume 1*, pages 42–48.

Regina Barzilay and Lillian Lee. 2002. Bootstrapping lexical choice via multiple-sequence alignment. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 164–171.

Anja Belz and Ehud Reiter. 2006. Comparing Automatic and Human Evaluation of NLG Systems. In *Proc. of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 313–320.

Anja Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 1:1–26.

Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon, and Hiroshi Shimodaira. 2005. Human-Computer Dialogue Simulation Using Hidden Markov Models. In *Proc. of ASRU*, pages 290–295.

Nina Dethlefs and Heriberto Cuayáhuitl. 2010. Hierarchical Reinforcement Learning for Adaptive Text Generation. *Proceeding of the 6th International Conference on Natural Language Generation (INLG)*.

Nina Dethlefs, Heriberto Cuayáhuitl, and Jette Viethen. 2011. Optimising Natural Language Generation Decision Making for Situated Dialogue. In *Proc. of the 12th Annual SIGdial Meeting on Discourse and Dialogue*.

Thomas G. Dietterich. 1999. Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition. *Journal of Artificial Intelligence Research*, 13:227–303.

Mary Ellen Foster and Jon Oberlander. 2006. Data-driven generation of emphatic facial displays. In *Proc. of the European Chapter of the Association for Computational Linguistic (EACL)*, pages 353–360.

Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Kristina Striegnitz. 2010. The GIVE-2 corpus of giving instructions in virtual environments. In *LREC*.

Michael A. K. Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. Longman, London.

Srinivasan Janarthanam and Oliver Lemon. 2010. Learning to adapt to unknown users: referring expression generation in spoken dialogue systems. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 69–78.

Alexander Koller, Kristina Striegnitz, Donna Byron, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2010. The first challenge on generating instructions in virtual environments. In M. Theune and E. Krahmer, editors, *Empirical Methods on Natural Language Generation*, pages 337–361, Berlin/Heidelberg, Germany. Springer.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 704–710.

W J M Levelt and S Kelter. 1982. Surface form and memory in question answering. *Cognitive Psychology*, 14.

François Mairesse, Milica Gašić, Filip Jurčíček, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2010. Phrase-based statistical language generation using graphical models and active learning. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1552–1561.

Alice H. Oh and Alexander I. Rudnicky. 2000. Stochastic language generation for spoken dialogue systems. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems - Volume 3*, pages 27–32.

Martin J. Pickering and Simon Garrod. 2004. Toward a mechanistc psychology of dialog. *Behavioral and Brain Sciences*, 27.

L R Rabiner. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of IEEE*, pages 257–286.

Verena Rieser, Oliver Lemon, and Xingkun Liu. 2010. Optimising information presentation for spoken dialogue systems. In *Proc. of the Annual Meeting of the Association for Computational Lingustics (ACL)*, pages 1009–1018.

Richard S Sutton and Andrew G Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA.

Menno van Zaanen. 2000. Bootstrapping syntax and recursion using alginment-based learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, pages 1063–1070, San Francisco, CA, USA.

Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. 1997. PARADISE: A framework for evaluating spoken dialogue agents. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 271–280.

Michael White. 2004. Reining in CCG chart realization. In *Proc. of the International Conference on Natural Language Generation (INLG)*, pages 182–191.