Comparing HMMs and Bayesian Networks for Surface Realisation

Nina Dethlefs
Heriot-Watt University
Edinburgh, Scotland
n.s.dethlefs@hw.ac.uk

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

Abstract

Natural Language Generation (NLG) systems often use a pipeline architecture for sequential decision making. Recent studies however have shown that treating NLG decisions jointly rather than in isolation can improve the overall performance of systems. We present a joint learning framework based on Hierarchical Reinforcement Learning (HRL) which uses graphical models for surface realisation. Our focus will be on a comparison of Bayesian Networks and HMMs in terms of user satisfaction and naturalness. While the former perform best in isolation, the latter present a scalable alternative within joint systems.

1 Introduction

NLG systems have traditionally used a pipeline architecture which divides the generation process into three distinct stages. **Content selection** chooses 'what to say' and constructs a semantic form. **Utterance planning** organises the message into submessages and **surface realisation** maps the semantics onto words. Recently, a number of studies have pointed out that many decisions made at these distinct stages require interrelated, rather than isolated, optimisations (Angeli et al., 2010; Lemon, 2011; Cuayáhuitl and Dethlefs, 2011a; Dethlefs and Cuayáhuitl, 2011a). The key feature of a joint architecture is that decisions of all three NLG stages share information and can be made in an interrelated fashion. We present a joint NLG framework based on Hierarchical RL and focus, in particular, on the surface realisation component of joint NLG systems. We compare the user satisfaction and naturalness of surface realisation using Hidden Markov Models (HMMs) and Bayesian Networks (BNs) which both have been suggested as generation spaces—spaces of surface form variants for a semantic concept—within joint NLG systems (Dethlefs and Cuayáhuitl, 2011a; Dethlefs and Cuayáhuitl, 2011b) and in isolation (Georgila et al., 2002; Mairesse et al., 2010).

2 Surface Realisation for Situated NLG

We address the generation of navigation instructions, where e.g. the semantic form $(path(target = end\_of\_corridor) \wedge (landmark = lift \wedge dir = left))$ can be expressed as 'Go to the end of the corridor', 'Head to the end of the corridor past the lift on your left' and many more. The best realisation depends on the space (types and properties of spatial objects), the user (position, orientation, prior knowledge) and decisions of content selection and utterance planning. These can be interrelated with surface realisation, for example:

(1) 'Follow this corridor and go past the lift on your left. Then turn right at the junction.'
(2) 'Pass the lift and turn right at the junction.'

Here, (1) is appropriate for a user unfamiliar with the space and a high information need, so that more information should be given. For a familiar user, however, who may know where the lift is, it is redundant and (2) is preferable, because it is more efficient. An unfamiliar user may get confused with just (2).

In this paper, we distinguish navigation of *destination* ('go back to the office'), *direction* ('turn left'), *orientation* ('turn around'), *path* ('follow the

2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 636–640,
Montréal, Canada, June 3-8, 2012. ©2012 Association for Computational Linguistics

corridor') and *straight'* ('go forward') in the GIVE corpus (Gargett et al., 2010). Users can react to an instruction by *performing the action*, *performing an undesired action*, *hesitating* or *requesting help*.

3 Jointly Learnt NLG: Hierarchical RL with Graphical Models

In a joint framework, each subtask of content selection, utterance planning and surface realisation has knowledge of the decisions made in the other two subtasks. In an isolated framework, this knowledge is absent. In the joint case, the relationship between hierarchical RL and graphical models is that the latter provide feedback to the former's surface realisation decisions according to a human corpus.

Hierarchical RL Our HRL agent consists of a hierarchy of discrete-time Semi-Markov Decision Processes, or SMDPs, M_j^i defined as 4-tuples $< S_j^i, A_j^i, T_j^i, R_j^i >$, where i and j uniquely identify a model in the hierarchy. These SMDPs represent generation subtasks, e.g. generating destination instructions. S_j^i is a set of states, A_j^i is a set of actions, and T_j^i is a probabilistic state transition function that determines the next state s' from the current state s and the performed action a. $R_j^i(s', \tau | s, a)$ is a reward function that specifies the reward that an agent receives for taking an action a in state s lasting τ time steps. Since actions in SMDPs may take a variable number of time steps to complete, the random variable τ represents this number of time steps. Actions can be either primitive or composite. The former yield single rewards, the latter correspond to SMDPs and yield cumulative rewards. The goal of each SMDP is to find an optimal policy π^* that maximises the reward for each visited state, according to $\pi^{*}{}_j^i(s) = \arg\max_{a \in A} Q^{*}{}_j^i(s, a)$, where $Q_j^i(s, a)$ specifies the expected cumulative reward for executing action a in state s and then following π^*. Please see (Dethlefs and Cuayáhuitl, 2011b) for details on the design of the hierarchical RL agent and the integration of graphical models for surface realisation.

Hidden Markov Models Representing surface realisation as an HMM can be roughly defined as the converse of POS tagging. While in POS tagging we map an observation string of words onto a hidden sequence of POS tags, in NLG we face the opposite scenario. Given an observation sequence of semantic symbols, we want to map it onto a hidden most likely sequence of words. We treat states as representing surface realisations for (observed) semantic classes, so that a sequence of states $s_0 ... s_n$ represents phrases or sentences. An observation sequence $o_0 ... o_n$ consists of a finite set of semantic symbols specific to an instruction type. Each symbol has an observation likelihood $b_s(o)_t$ giving the probability of observing o in state s at time t. We created the HMMs and trained the transition and emission probabilities from the GIVE corpus using the Baum-Welch algorithm. Please see Fig. 1 for an example HMM and (Dethlefs and Cuayáhuitl, 2011a) for details on using HMMs for surface realisation.

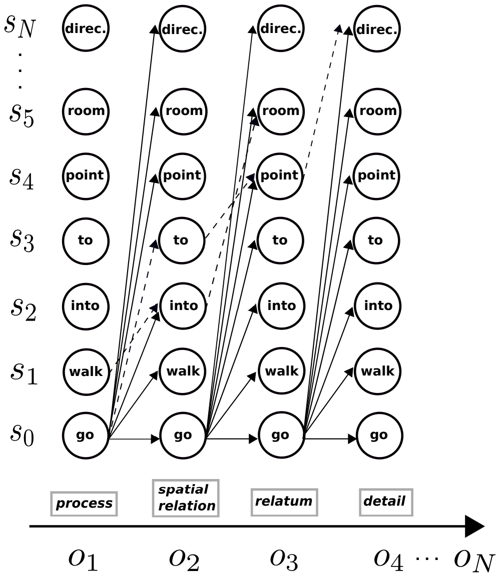

Figure 1: Example trellis for an HMM for destination instructions (not all states and transitions are shown). Dashed arrows show paths that occur in the corpus.

Bayesian Networks Representing a surface realiser as a BN, we can model the dynamics between semantic concepts and their realisations. A BN models a joint probability distribution over a set of random variables and their dependencies based on a directed acyclic graph, where each node represents a variable Y_j with parents $pa(Y_j)$. Due to the Markov condition, each variable depends only on its parents, resulting in a unique joint probability distribution $p(Y) = \Pi p(Y_j | pa(Y_j))$, where every variable is associated with a conditional probability distribution

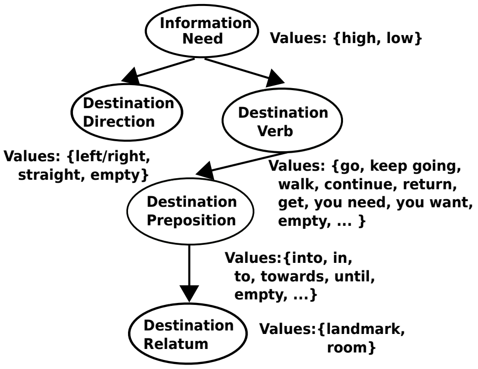
Information Need
Values: {high, low}
Destination Direction
Destination Verb
Values: {left/right, straight, empty}
Values: {go, keep going, walk, continue, return, get, you need, you want, empty, ... }
Destination Preposition
Values:{into, in, to, towards, until, empty, ...}
Destination Relatum
Values:{landmark, room}

Figure 2: BN for generating destination instructions.

$p(Y_j|pa(Y_j))$. The meaning of random variables corresponds to semantic symbols. The values of random variables correspond to surface variants of a semantic symbol. Figure 2 shows an example BN with two main dependencies. First, the random variable 'information need' influences the inclusion of optional semantic constituents and the process of the utterance ('destination verb'). Second, a sequence of dependencies spans from the verb to the end of the utterance ('destination relatum'). The first dependency is based on the intuition that more detail is needed in an instruction for users with high information need (e.g. with little prior knowledge).[1] The second dependency is based on the hypothesis that the value of one constituent can be estimated based on the previous constituent. In the future, we may compare different configurations and effects of word order. Given the word sequence represented by lexical and syntactic variables $Y_0 ... Y_n$, and situation-based variables $Y_{n+1} ... Y_m$, we can compute the posterior probability of a random variable Y_j. The parameters of the BNs were estimated using MLE. Please see (Dethlefs and Cuayáhuitl, 2011b) for details on using BNs for surface realisation within a joint learning framework.

4 Experimental Setting

We compare instructions generated with the HMMs and BNs according to their *user satisfaction* and their *naturalness*. The learning agent is trained using the reward function $Reward = User\_satisfaction \times P(w_0 \ldots w_n) \times CAS$.[2] $User\_satisfaction$ is a function of task success and the number of user turns based on the PARADISE framework[3] (Walker et al., 1997) and CAS refers to the proportion of repetition and variation in surface forms. Our focus in this short paper is on $P(w_0 \ldots w_n)$ which rewards the agent for having generated a surface form sequence $w_0 \ldots w_n$. In HMMs, this corresponds to the forward probability—obtained from the Forward algorithm—of observing the sequence in the data. In BNs, $P(w_0 \ldots w_n)$ corresponds to $P(Y_j = v_x|pa(Y_j) = v_y)$, the posterior probability given the chosen values v_x and v_y of random variables and their dependencies. We assign a reward of -1 for each action to prevent loops.

5 Experimental Results

User satisfaction Our trained policies learn the same content selection and utterance planning behaviour reported by (Dethlefs and Cuayáhuitl, 2011b). These policies contribute to the *user satisfaction* of instructions. BNs and HMMs however differ in their surface realisation choices. Figure 3 shows the performance in terms of average rewards over time for both models within the joint learning framework and in isolation.[4] For ease of comparison, a learning curve using a *greedy* policy is also shown. It always chooses the most likely surface form according to the human corpus without taking other tradeoffs into account. Within the joint framework, both BNs and HMMs learn to generate context-sensitive surface forms that balance the tradeoffs of the most likely sequence (according to the human corpus) and the one that best corresponds to the user's information need (e.g., using nick names of rooms for familiar users). The BNs

[1] This is key to the joint treatment of content selection and surface realisation: if an utterance is not informative in terms of content, it will receive bad rewards, even with good surface realisation choices (and vice versa).

[2] This reward function, the simulated environment and training parameters were adapted from (Dethlefs and Cuayáhuitl, 2011b) to allow a comparison with related work in using graphical models for surface realisation. Simulation is based on uni- and bigrams for the spatial setting and Naive Bayes Classification for user reactions to system instructions.

[3] See (Dethlefs et al., 2010) for evidence of the correlation between user satisfaction, task success and dialogue length.

[4] In the isolated case, subtasks of content selection, utterance planning and surface realisation are blind regarding the decisions made by other subtasks, but in the joint case they are not.

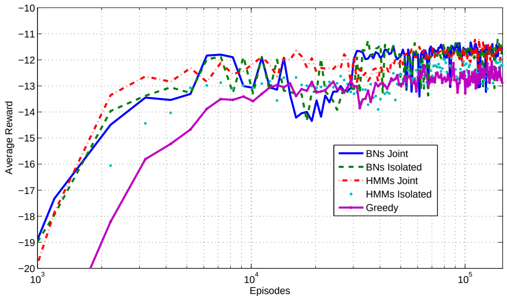

Figure 3: Performance of HMMs, BNs and a *greedy* baseline in conjunction and isolation of the joint framework.

reach an average reward[5] of -11.53 and outperform the HMMs (average -11.64) only marginally by less than one percent. BNs and HMMs improve the *greedy* baseline by 6% ($p < 0.0001$, $r = 0.90$). While BNs reach the same performance in isolation of the joint framework, the performance of HMMs deteriorates significantly to an average reward of -12.12. This corresponds to a drop of 5% ($p < 0.0001$, $r = 0.79$) and is nearly as low as the *greedy* baseline. HMMs thus reach a comparable performance to BNs as a result of the joint learning architecture: the HRL agent will discover the non-optimal behaviour that is caused by the HMM's lack of context-awareness (due to their independence assumptions) and learn to balance this drawback by learning a more comprehensive policy itself. For the more context-aware BNs this is not necessary.

Naturalness We compare the instructions generated with HMMs and BNs regarding their human-likeness based on the Kullback-Leibler (KL) divergence. It computes the difference between two probability distributions. For evidence of its usefulness for measuring naturalness, cf. (Cuayáhuitl, 2010). We compare human instructions (based on strings) drawn from the corpus against strings generated by the HMMs and BNs to see how similar both are to human authors. Splitting the human instructions in half and comparing them to each other indicates how similar human authors are to each other. It yields a KL score of 1.77 as a gold standard (the lower the better). BNs compared with human data obtain a score of 2.83 and HMMs of 2.80. The difference in terms of similarity with humans for HMMs and BNs in a joint NLG model is not significant.

Discussion While HMMs reach comparable user satisfaction and naturalness to BNs in a joint system, they show a 5% lower performance in isolation. This is likely caused by their conditional independence assumptions: (a) the Markov assumption, (b) the stationary assumption, and (c) the observation independence assumption. Even though these can make HMMs easier to train and scale than more structured models such as BNs, it also puts them in a disadvantage concerning context-awareness and accuracy as shown by our results. In contrast, the random variables of BNs allow them to keep a structured model of the space, user, and relevant content selection and utterance planning choices. BNs are thus able to compute the posterior probability of a surface form based on all relevant properties of the current situation (not just the occurrence in a corpus). While BNs also place independence assumptions on their variables, they usually overcome the problem of lacking context-awareness by their dependencies across random variables. However, BNs also face limitations. Given the dependencies they postulate, they are typically more data intensive and less scalable than less structured models such as HMMs. This can be problematic for large domains such as many real world applications. Regarding their application to surface realisation, we can argue that while BNs are the best performing model in isolation, HMMs represent a cheap and scalable alternative especially for large-scale problems in a joint NLG system.

6 Conclusion and Future Work

We have compared the user satisfaction and naturalness of instructions generated with HMMs and BNs in a joint HRL model for NLG. Results showed that while BNs perform best in isolation, HMMs represent a cheap and scalable alternative within the joint framework. This is particularly attractive for large-scale, data-intensive systems. While this paper has focused on instruction generation, the hierarchical approach in our learning framework helps to scale up to larger NLG tasks, such as text or paragraph generation. Future work could test this claim, compare other graphical models, such as dynamic BNs, and aim for a comprehensive human evaluation.

[5] The average rewards of agents have negative values due to the negative reward of -1 the agent receives for each action.

Acknowledgements This research was funded by the European Commission's FP7 programmes under grant agreement no. 287615 (PARLANCE) and no. ICT-248116 (ALIZ-E).

References

Angeli, G., Liang, P. and D. Klein (2010). A simple domain-independent probabilistic approach to generation , *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* .

Cuayáhuitl, H., Renals, S., Lemon, O. and H. Shimodaira (2010). Evaluation of a Hierarchical Reinforcement Learning Spoken Dialogue System, *Computer Speech and Language 24*.

Cuayáhuitl, H., and N. Dethlefs (2011a). Spatially-Aware Dialogue Control Using Hierarchical Reinforcement Learning, *ACM Transactions on Speech and Language Processing (Special Issue on Machine Learning for Robust and Adaptive Spoken Dialogue Systems 7(3)*.

Dethlefs, N. and H. Cuayáhuitl, 2011. Hierarchical Reinforcement Learning and Hidden Markov Models for Task-Oriented Natural Language Generation, *In Proc. of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-HLT)*.

Dethlefs, N. and H. Cuayáhuitl, 2011. 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)*.

Dethlefs, N., Cuayáhuitl, H., Richter, K.-F., Andonova, E. and J. Bateman, 2010. Evaluating Task Success in a Dialogue System for Indoor Navigation, *In Proceedings of the Workshop on the Semantics and Pragmatics of Dialogue (SemDial)*.

Gargett, A., Garoufi, K., Koller, A. and K. Striegnitz (2010). The GIVE-2 Corpus of Giving Instructions in Virtual Environments, *Proc. of the 7th International Conference on Language Resources and Evaluation*.

Georgila, K., Fakotakis, N. and Kokkinakis, G. (2002). Stochastic Language Modelling for Recognition and Generation in Dialogue Systems. *TAL (Traitement automatique des langues) Journal*, Vol. 43(3).

Lemon, O. (2011). Learning what to say and how to say it: joint optimization of spoken dialogue management and Natural Language Generation, *Computer Speech and Language 25(2)*.

Mairesse, F., Gašić, M., Jurčíček, F., Keizer, S., Thomson, B., Yu, K. and S. Young (2010). Phrase-based statistical language generation using graphical models and active learning, *Proc. of the 48th Annual Meeting of the Association for Computational Linguistics*.

Walker, M., Litman, D., Kamm, C. and A. Abella (1997). PARADISE: A Framework for Evaluating Spoken Dialogue Agents, *Proceedings of the Annual Meeting of the Association for Computational Linguistic (ACL)*.