Conditional Random Fields for Responsive Surface Realisation using Global Features

Nina Dethlefs, Helen Hastie, Heriberto Cuayáhuitl and Oliver Lemon
Mathematical and Computer Sciences
Heriot-Watt University, Edinburgh
n.s.dethlefs | h.hastie | h.cuayahuitl | o.lemon@hw.ac.uk

Abstract

Surface realisers in spoken dialogue systems need to be more responsive than conventional surface realisers. They need to be sensitive to the utterance context as well as robust to partial or changing generator inputs. We formulate surface realisation as a sequence labelling task and combine the use of conditional random fields (CRFs) with semantic trees. Due to their extended notion of context, CRFs are able to take the global utterance context into account and are less constrained by local features than other realisers. This leads to more natural and less repetitive surface realisation. It also allows generation from partial and modified inputs and is therefore applicable to incremental surface realisation. Results from a human rating study confirm that users are sensitive to this extended notion of context and assign ratings that are significantly higher (up to 14%) than those for taking only local context into account.

1 Introduction

Surface realisation typically aims to produce output that is grammatically well-formed, natural and cohesive. Cohesion can be characterised by lexical or syntactic cues such as repetitions, substitutions, ellipses, or connectives. In automatic language generation, such properties can sometimes be difficult to model, because they require rich context-awareness that keeps track of all (or much) of what was generated before, i.e. a growing generation history. In text generation, cohesion can span over the entire text. In interactive settings such as generation within a spoken dialogue system (SDS), a challenge is often to keep track of cohesion over several utterances. In addition, since interactions are dynamic, generator inputs from the dialogue manager can sometimes be partial or subject to subsequent modification. This has been addressed by work on incremental processing (Schlangen and Skantze, 2009). Since dialogue acts are passed on to the generation module as soon as possible, this can sometimes lead to incomplete generator inputs (because the user is still speaking), or inputs that are subject to later modification (because of an initial ASR mis-recognition).

In this paper, we propose to formulate surface realisation as a sequence labelling task. We use conditional random fields (Lafferty et al., 2001; Sutton and McCallum, 2006), which are suitable for modelling rich contexts, in combination with semantic trees for rich linguistic information. This combination is able to keep track of dependencies between syntactic, semantic and lexical features across multiple utterances. Our model can be trained from minimally labelled data, which reduces development time and may (in the future) facilitate an application to new domains.

The domain used in this paper is a pedestrian walking around a city looking for information and recommendations for local restaurants from an SDS. We describe here the module for surface realisation. Our main hypothesis is that the use of global context in a CRF with semantic trees can lead to surface realisations that are better phrased, more natural and less repetitive than taking only local features into account. Results from a human rating study confirm this hypothesis. In addition, we compare our system with alternative surface realisation methods from the literature, namely, a rank and boost approach and n-grams.

Finally, we argue that our approach lends itself

to surface realisation within incremental systems, because CRFs are able to model context across full as well as partial generator inputs which may undergo modifications during generation. As a demonstration, we apply our model to incremental surface realisation in a proof-of-concept study.

2 Related Work

Our approach is most closely related to Lu et al. (2009) who also use CRFs to find the best surface realisation from a semantic tree. They conclude from an automatic evaluation that using CRF-based generation which takes long-range dependencies into account outperforms several baselines. However, Lu et al.'s generator does not take context beyond the current utterance into account and is thus restricted to local features. Furthermore, their model is not able to modify generation results on the fly due to new or updated inputs.

In terms of surface realisation from graphical models (and within the context of SDSs), our approach is also related to work by Georgila et al. (2002) and Dethlefs and Cuayáhuitl (2011b), who use HMMs, Dethlefs and Cuayáhuitl (2011a) who use Bayes Nets, and Mairesse et al. (2010) who use Dynamic Bayes Nets within an Active Learning framework. The last approach is also concerned with generating restaurant recommendations within an SDS. Specifically, their system optimises its performance online, during the interaction, by asking users to provide it with new textual descriptions of concepts, for which it is unsure of the best realisation. In contrast to these related approaches, we use undirected graphical models which are useful when the natural directionality between the input variables is unknown.

In terms of surface realisation for SDSs, Oh and Rudnicky (2000) present foundational work in using an n-gram-based system. They train a surface realiser based on a domain-dependent language model and use an overgeneration and ranking approach. Candidate utterances are ranked according to a penalty function which penalises too long or short utterances, repetitious utterances and utterances which either contain more or less information than required by the dialogue act. While their approach is fast to execute, it has the disadvantage of not being able to model long-range dependencies. They show that humans rank their output equivalently to template-based generation.

Further, our approach is related to the SPaRKy sentence generator (Walker et al., 2007). SPaRKy was also developed for the domain of restaurant recommendations and was shown to be equivalent to or better than a carefully designed template-based generator which had received high human ratings in the past (Stent et al., 2002). It generates sentences in two steps. First, it produces a randomised set of alternative realisations, which are then ranked according to a mapping from sentence plans to predicted human ratings using a boosting algorithm. As in our approach, SPaRKy distinguishes local and global features. Local features take only information of the current tree node into account, including its parents, siblings and children, while global features take information of the entire utterance into account. While SPaRKy is shown to reach high output quality in comparison to a template-based baseline, the authors acknowledge that generation with SPaRKy is rather slow when applied in a real-time SDS. This could present a problem in incremental settings, where generation speed is of particular importance.

The SPaRKy system is also used by Rieser et al. (2011), who focus on information presentation strategies for restaurant recommendations, summaries or comparisons within an SDS. Their surface realiser is informed by the highest ranked SPaRKy outputs for a particular information presentation strategy and will constitute one of our baselines in the evaluation.

More work on trainable realisation for SDSs generally includes Bulyko and Ostendorf (2002) who use finite state transducers, Nakatsu and White (2006) who use supervised learning, Varges (2006) who uses chart generation, and Konstas and Lapata (2012) who use weighted hypergraphs, among others.

3 Cohesion across Utterances

3.1 Tree-based Semantic Representations

The restaurant recommendations we generate can include any of the attributes shown in Table 1. It is then the task of the surface realiser to find the best realisation, including whether to present them in one or several sentences. This often is a sentence planning decision, but in our approach it is handled using CRF-based surface realisation. The semantic forms underlying surface realisation can be produced in many ways. In our case, they are produced by a reinforcement learning agent which orders semantic attributes in the tree ac-

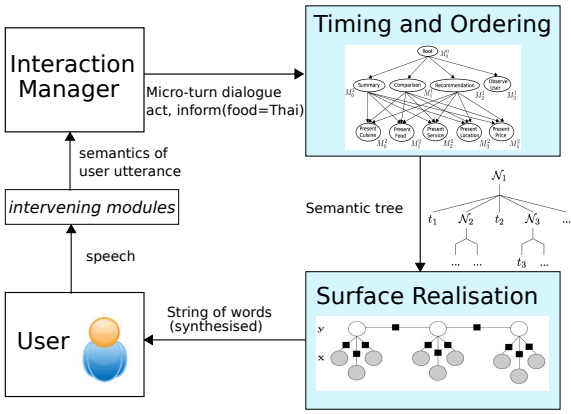

Figure 1: Architecture of our SDS with a focus on the NLG components. While the user is speaking, the dialogue manager sends dialogue acts to the NLG module, which uses reinforcement learning to order semantic attributes and produce a semantic tree (see Dethlefs et al. (2012b)). This paper focuses on surface realisation from these trees using a CRF as shown in the surface realisation module.

Slot	Example
ADDRESS	The venue's address is ...
AREA	It is located in ...
FOOD	The restaurant serves ... cuisine.
NAME	The restaurant's name is ...
PHONE	The venue's phone number is ...
POSTCODE	The postcode is ...
QUALITY	This is a ... venue.
PRICE	It is located in the ... price range.
SIGNATURE	The venue specialises in ...
VENUE	This venue is a ...

Table 1: Semantic slots required for our domain along with example realisations. Attributes can be combined in all possible ways during generation.

cording to their confidence in the dialogue. This is because SDSs can often have uncertainties with regard to the user's actual desired attribute values due to speech recognition inaccuracies. We therefore model all semantic slots as probability distributions, such as *inform(food=Indian, 0.6) or inform(food=Italian, 0.4)* and apply reinforcement learning to finding the optimal sequence for presentation. Please see Dethlefs et al. (2012b) for details. Here, we simply assume that a semantic form has been produced by a previous processing module.

As shown in the architecture diagram in Figure 1, a CRF surface realiser takes a semantic tree as input. We represent these as context-free trees which can be defined formally as 4-tuples

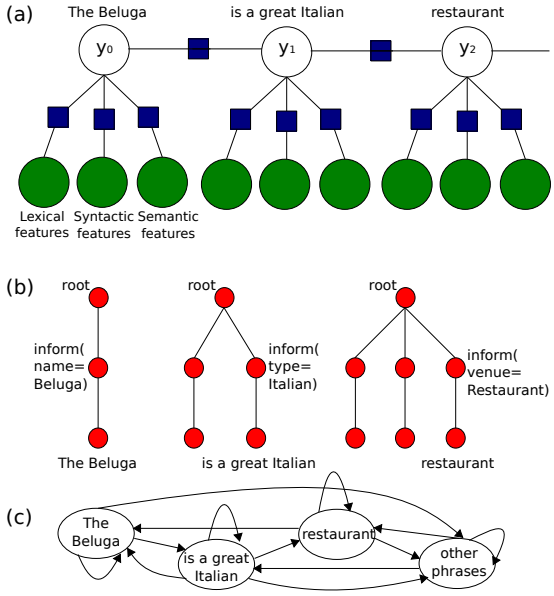

Figure 2: (a) Graphical representation of a linear-chain Conditional Random Field (CRF), where empty nodes correspond to the labelled sequence, shaded nodes to linguistic observations, and dark squares to feature functions between states and observations; (b) Example semantic trees that are updated at each time step in order to provide linguistic features to the CRF (only one possible surface realisation is shown and parse categories are omitted for brevity); (c) Finite state machine of phrases (labels) for this example.

$\{S, T, N, H\}$, where S is a start symbol, typically the root node of the tree; $T = \{t_0, t_1, t_2 \dots t_{|T|}\}$ is a set of terminal symbols, corresponding to single phrases; $N = \{n_0, n_1, n_2 \dots n_{|N|}\}$ is a set of non-terminal symbols corresponding to semantic categories, and $H = \{h_0, h_1, h_2 \dots h_{|H|}\}$ is a set of production rules of the form $n \rightarrow \alpha$, where $n \in N$, $\alpha \in T \cup N$. The production rules represent alternatives at each branching node where the CRF is consulted for the best available expansion from the subset of possible ones. All nodes in the tree are annotated with a semantic concept (obtained from the semantic form) as well as their parse category.

3.2 Conditional Random Fields for Phrase-Based Surface Realisation

The main idea of our approach is to treat surface realisation as a sequence labelling task in which a sequence of semantic inputs needs to be labelled with appropriate surface realisations. The task is therefore to find a mapping between (observed)

lexical, syntactic and semantic features and a (hidden) best surface realisation.

We use the linear-chain Conditional Random Field (CRF) model for statistical phrase-based surface realisation, see Figure 2 (a). This probabilistic model defines the posterior probability of labels (surface realisation phrases) $\mathbf{y}=\{y_1,\ldots,y_{|\mathbf{y}|}\}$ given features $\mathbf{x}=\{x_1,\ldots,x_{|\mathbf{x}|}\}$ (informed by a semantic tree, see Figure 2 (b)), as

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\prod_{t=1}^{T}\exp\left\{\sum_{k=1}^{K}\theta_k\Phi_k(y_t,y_{t-1},\mathbf{x}_t)\right\},$$

where $Z(\mathbf{x})$ is a normalisation factor over all possible realisations (i.e. labellings) of $\mathbf{x}$ such that the sum of all terms is one. The parameters θ_k are weights corresponding to feature functions $\Phi_k(.)$, which are real values describing the label state y at time t based on the previous label state y_{t-1} and features $\mathbf{x}_t$. For example: from Figure 2 (c), Φ_k might have the value $\Phi_k = 1.0$ for the transition from *"The Beluga"* to *"is a great Italian"*, and 0.0 elsewhere. The parameters θ_k are set to maximise the conditional likelihood of phrase sequences in the training data set. They are estimated using the gradient ascent algorithm.

After training, labels can be predicted for new sequences of observations. The most likely phrase sequence is expressed as

$$\mathbf{y}^{*} = \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}),$$

which is computed using the Viterbi algorithm. We use the Mallet package[1] (McCallum, 2002) for parameter learning and inference.

3.3 Feature Selection and Training

The following features define the generation context used during training of the CRF. The generation context includes everything that has been generated for the current utterance so far. All features can be obtained from a semantic input tree.

- Lexical items of parents and siblings,
- Semantic types in expansion,
- Semantic types of parents and siblings,
- Parse category of expansion,
- Parse categories of parents and siblings.

We use the StanfordParser[2] (Marneffe et al., 2006) to obtain the parse category for each tree node. The semantics for each node are derived from the input dialogue acts (these are listed in Table 1) and are associated with nodes. The lexical items are present in the generation context and are mapped to semantic tree nodes.

As an example, for generating an utterance (label sequence) such as *The Beluga is a great restaurant. It is located in the city centre.*, each generation step needs to take the features of the entire generation history into account. This includes all individual lexical items generated, the semantic types used and the parse categories for each tree node involved. For the first constituent, *The Beluga*, this corresponds to the features {ˆ BEGIN NAME} indicating the beginning of a sentence (where empty features are omitted), the beginning of a new generation context and the next semantic slot required. For the second constituent, *is a great restaurant*, the features are {THE BELUGA NAME NP VENUE}, i.e. including the generation history (with lexical items and parse category added for the first constituent) and the semantics of the next required slot, VENUE. In this way, a sequence of surface form constituents is generated corresponding to latent states in the CRF.

Since global utterance features capture the full generation context (i.e. beyond the current utterance), we are also able to model phenomena such as co-references and pronouns. This is useful for longer restaurant recommendations which may span over more than one utterance. If the generation history already contains a semantic attribute, e.g. the restaurant name, the CRF may afterwards choose a pronoun, e.g. *it*, which has a higher likelihood than using the proper name again. Similarly, the CRF may decide to realise a new attribute as constituents of different order, such as a sentence or PP, depending on the length, number and parse categories of previously generated output. In this way, our approach implicitly treats sentence planning decisions such as the distribution of content over a set of messages in the same way as (or as part of) surface realisation. A further capability of our surface realiser is that it can generate complete phrases from full as well as partial dialogue acts. This is useful in interactive contexts, where we need as much robustness as possible. A demonstration of this is given in Section 5 in an application to incremental surface realisation.

To train the CRF, we used a data set of 552 restaurant recommendations from the website The

[1] http://mallet.cs.umass.edu/

[2] http://nlp.stanford.edu/software/lex-parser.shtml

List.[3] The data contains recommendations such as *Located in the city centre, Beluga is a stylish yet laid-back restaurant with a smart menu of modern European cuisine*.

3.4 Grammar Induction

The grammar g of surface realisation candidates is obtained through an automatic grammar induction algorithm which can be run on unlabelled data and requires only minimal human intervention. This grammar defines the surface realisation space for the CRFs. We provide the human corpus of restaurant recommendations from Section 3.3 as input to grammar induction. The algorithm is shown in Algorithm 1. It first identifies all semantic attributes of interest in an utterance, in our case those specified in Table 1, and replaces them by a variable. These attributes include food types, such as *Mexican, Chinese*, particular parts of town, prices, etc. About 45% of them can be identified based on heuristics. The remainder needs to be hand-annotated at the moment, which includes mainly attributes like restaurant names or quality attributes, such as *delicate, exquisite*, etc. Subsequently, all utterances are parsed using the Stanford parser to obtain constituents and are integrated into the grammar under construction. The non-terminal symbols are named after the automatically annotated semantic attributes contained in their expansion, e.g. NAME_QUALITY $\rightarrow$ *The* $\$name\$$ *is of* $\$quality\$$ *quality.* In this way, each non-terminal symbol has a semantic representation and an associated parse category. In total, our induced grammar contains more than 800 rules.

4 Evaluation

To evaluate our approach, we focus on a subjective human rating study which aims to determine whether CRF-based surface realisation that takes the full generation context into account, called **CRF (global)**, is perceived better by human judges than one that uses a CRF but just takes local context into account, called **CRF (local)**. While CRF (global) uses features from the entire generation history, CRF (local) uses only features from the current tree branch. We assume that cohesion can be identified by untrained judges as natural, well-phrased and non-repetitive surface forms. To examine differences in methodology between

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

Algorithm 1 Grammar Induction.

function FINDGRAMMAR(utterances u, semantic attributes a) **return** $grammar$
  for each utterance u **do**
    if u contains a semantic attribute from a, such as $venue$, $cuisine$, etc. **then**
      Find and replace the attribute by its semantic variable, e.g. $\$venue\$$.
    end if
    Parse the sentence and induce a set of rules $\alpha \rightarrow \beta$, where α is a semantic variable and β is its parse.
    Traverse the parse tree in a top-down, depth-first search and
    if expansion β exists **then**
      continue
    else if non-terminal α exists **then**
      add new expansion β to α.
    else write new rule $\alpha \rightarrow \beta$.
    end if
    Write grammar.
  end for
end function

CRFs and other state-of-the-art methods, we also compare our system to two other baselines:

- **CLASSiC** corresponds to the system reported in Rieser et al. (2011),[4] which generates restaurant recommendations based on the SPaRKy system (Walker et al., 2007), and has received high ratings in the past. SPaRKy uses global utterance features.
- n**-grams** represents a simple 5-gram baseline that is similar to Oh and Rudnicky (2000)'s system. We will sample from the most likely slot realisations that do not contain a repetition and include exactly the required slot values. Local context only is taken into account.

4.1 Human Rating Study

We carried out a user rating study on the CrowdFlower crowd sourcing platform.[5] Each participant was shown part of a real human-system dialogue that emerged as part of the CLASSiC project evaluation (Rieser et al., 2011). All dialogues and data are freely available from `http://www.classic-project.org`. Each dialogue contained two variations for one of the utterances. These variations were generated from two out of the four systems described above. The order that these were presented to the participant was counterbalanced. Table 2 gives an example of a dialogue segment presented to the participants.

[4] In Rieser et al. (2011), this system is referred to as the TIP system, which generates summaries, comparisons or recommendations for restaurants. For the present study, we com-

SYS	Thank you for calling the Cambridge Information system. Your call will be recorded for research purposes. You may ask for information about a place to eat, such as a restaurant, a pub, or a cafe. How may I help you?
USR	I want to find an American restaurant which is in the very expensive area.
SYS A	*The restaurant Gourmet Burger is an outstanding, expensive restaurant located in the central area.*
SYS B	*Gourmet Burger is a smart and welcoming restaurant. Gourmet Burger provides an expensive dining experience with great food and friendly service. If you're looking for a central meal at an expensive price.*
USR	What is the address and phone number?
SYS	Gourmet Burger is on Regent Street and its phone number is 01223 312598.
USR	Thank you. Good bye.

Table 2: Example dialogue for participants to compare alternative outputs in italics, USR=user, SYS A=CRF (global), SYS B=CRF(local).

System	Natural	Phrasing	Repetit.
CRF global	**3.65**	**3.64**	**3.65**
CRF local	3.10*	3.19*	3.13*
CLASSiC	3.53*	3.59	3.48*
n-grams	3.01*	3.09*	3.32*

Table 3: Subjective user ratings. Significance with CRF (global) at $p<0.05$ is indicated as *.

44 participants gave a total of 1,830 ratings of utterances produced across the four systems. Fluent speakers of English only were requested and the participants were from the United States. They were asked to rate each utterance on a 5 point Likert scale in response to the following questions (where 5 corresponds to *totally agree* and 1 corresponds to *totally disagree*):

- The utterance was natural, i.e. it could have been produced by a human. (*Natural*)
- The utterance was phrased well. (*Phrasing*)
- The utterance was repetitive. (*Repetitive*)

4.2 Results

We can see from Table 3 that across all the categories, the CRF (global) gets the highest overall ratings. This difference is significant for all categories compared with CRF (local) and n-grams (using a 1-sided Mann Whitney U-test, $p < 0.001$). Possibly this is because the local context taken into account by both systems was not enough to ensure cohesion across surface phrases. It is not possible, e.g., to cover co-references within a local context only or discourse markers that refer beyond the current utterance. This can lead to short and repetitive phrases, such as *Make your way to Gourmet Burger. The food quality is outstanding. The prices are expensive.* generated by the n-gram baseline.

The CLASSiC baseline, based on SPaRKy, was the most competitive system in our comparison. None-the-less CRF (global) is rated higher across categories and significantly so for *Natural* ($p < 0.05$) and *Repetitive* ($p < 0.005$). For *Phrasing*, there is a trend but not a significant difference ($p < 0.16$). All comparisons are based on a 1-sided Mann Whitney U-test. A qualitative comparison between the CRF (global) and CLASSiC outputs showed the following. CLASSiC utterances tend to be longer and contain more sentences than CRF (global) utterances. While CRF (global) often decides to aggregate attributes into one sentence, such as *the Beluga is an outstanding restaurant in the city centre*, CLASSiC tends to rely more on individual messages, such as *The Beluga is an outstanding restaurant. It is located in the city centre.* A possible reason is that while CRF (global) is able to take features beyond an utterance into account, CLASSiC/SPaRKy is restricted to global features of the current utterance.

We can further compare our results with Rieser et al. (2011) and Mairesse et al. (2010) who also generate restaurant recommendations and asked similar questions to participants as we did. Rieser et al. (2011)'s system received an average rating of 3.58[6] in terms of *Phrasing* which compares to our 3.64. This difference is not significant, and in line with the user ratings we observed for the CLASSiC system above (3.59). Mairesse et al. (2010) achieved an average score of 4.05 in terms of *Natural* in comparison to our 3.65. This difference is significant at $p<0.05$. Possibly their better performance is due to the data set being more "in domain" than ours. They collected data from humans that was written specifically for the task that the system was tested on. In contrast, our system was trained on freely available data that was written by professional restaurant reviewers. Unfortunately, we cannot compare across other categories,

pare only with the subset of recommendations.

[5] `http://www.crowdflower.com`

[6] This was rescaled from a 1-6 scale.

USR1	I'm looking for a nice restaurant in the centre.
SYS1	*inform(area=centre [0.2], food=Thai [0.3])* *inform(name=Bangkok [0.3])* So you're looking for a Thai ...
USR2	*[barges in]* No, I'm looking for a restaurant with good quality food.
SYS2	*inform(quality=good [0.6], name=Beluga [0.6])* Oh sorry, so a nice restaurant located ...
USR3	*[barges in]* ... in the city centre.
SYS3	*inform(area=centre [0.8])*

Table 4: Example dialogue where the dialogue manager needs to send incremental updates to the NLG. Incremental surface realisation from semantic trees for this dialogue is shown in Figure 3.

because the authors tested only for *Phrasing* and *Natural*, respectively.

5 Incremental Surface Realisation

Recent years have seen increased interest in incremental dialogue processing (Skantze and Schlangen, 2009; Schlangen and Skantze, 2009). The main characteristic of incremental architectures is that instead of waiting for the end of a user turn, they begin to process the input stream as soon as possible, updating their processing hypotheses as more information becomes available. From a dialogue perspective, they can be said to work on partial rather than full dialogue acts.

With respect to surface realisation, incremental NLG systems have predominantly relied on pre-defined templates (Purver and Otsuka, 2003; Skantze and Hjalmarsson, 2010; Dethlefs et al., 2012a), which limits the flexibility and quality of output generation. Buschmeier et al. (2012) have presented a system which systematically takes the user's acoustic understanding problems into account by pausing, repeating or re-phrasing if necessary. Their approach is based on SPUD (Stone et al., 2003), a constraint satisfaction-based NLG architecture and marks important progress towards more flexible incremental surface realisation. However, given the human labour involved in constraint specification, cohesion is often limited to a local context. Especially for long utterances or such that are separated by user turns, this may lead to surface form increments that are not well connected and lack cohesion.

5.1 Application to Incremental SR

This section will discuss a proof-of-concept application of our approach to incremental surface realisation. Table 4 shows an example dialogue between a user and system that contains a number of incremental phenomena that require hypothesis updates, system corrections and user barge-ins. Incremental surface realisation for this dialogue is shown in Figure 3, where processing steps are indicated as bold-face numbers and are triggered by partial dialogue acts that are sent from the dialogue manager, such as *inform(area=centre [0.2])*. The numbers in square brackets indicate the system's confidence in the attribute-value pair. Once a dialogue act is observed by the NLG system, a reinforcement learning agent determines the order of attributes and produces a semantic tree, as described in Section 3.1. Since the semantic forms are constructed incrementally, new tree nodes can be attached to and deleted from an existing tree, depending on what kind of update is required.

In the dialogue in Table 4, the user first asks for a *nice restaurant in the centre*. The dialogue manager constructs a first attribute-value slot, *inform(area=centre [0.2], ...)*, and passes it on to NLG.[7] In Figure 3, we can observe the corresponding NLG action, a first tree is created with just a root node and a node representing the area slot (step 1). In a second step, the semantically annotated node gets **expanded** into a surface form that is chosen from a set of candidates (shown in curly brackets). The CRF is responsible for this last step. Since there is no preceding utterance, the best surface form is chosen based on the semantics alone. Active tree nodes, i.e. those currently under generation, are indicated as asterisks in Figure 3. Currently inactive nodes are shown as circles.

Step 3 then further expands the current tree adding a node for the food type and the name of a restaurant that the dialogue manager had passed. We see here that attributes can either be primitive or complex. Primitive attributes contain a single semantic type, such as *area*, whereas complex attributes contain multiple types, such as *food, name* and need to be decomposed in a later processing step (see steps 4 and 6). Step 5 again uses the CRF

[7] Note here that the information passed on to the NLG is distinct from the dialogue manager's own actions. In the example, the NLG is asked to generate a recommendation, but the dialogue manager actually decides to clarify the user's preferences due to low confidence. This scenario is an example of generator inputs that may get revised afterwards.

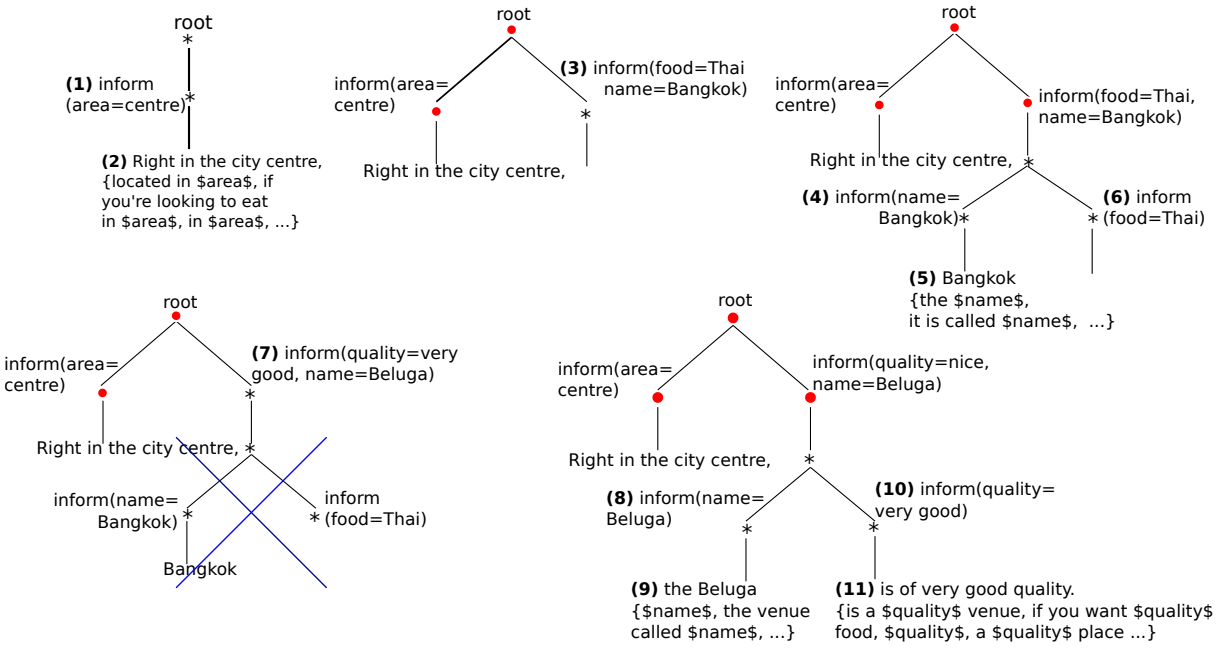

Figure 3: Example of incremental surface realisation, where each generation step is indicated by a number. Active generation nodes are shown as asterisks and deletions are shown as crossed out. Lexical and semantic features are associated with their respective nodes. Syntactic information in the form of parse categories are also taken into account for surface realisation, but have been omitted in this figure.

to obtain the next surface realisation that connects with the previous one (so that a sequence of realisation "labels" appears: *Right in the city centre* and *Bangkok*). It takes the full generation context into account to ensure a globally optimal choice. This is important, because the local context would otherwise be restricted to a partial dialogue act, which can be much smaller than a full dialogue act and thus lead to short, repetitive sentences.

The dialogue continues as the system implicitly confirms the user's preferred restaurant (SYS1). At this point, we encounter a user barge-in correcting the desired choice. As a consequence, the dialogue manager needs to **update** its initial hypotheses and communicate this to NLG. Here, the last three tree nodes need to be **deleted** from the tree because the information is no longer valid. This update and the deletion is shown in step 7. Afterwards, the dialogue continues and NLG involves mainly expanding the current tree into a full sequence of surface realisations for partial dialogue acts which come together into a full utterance.

This example illustrates three incremental processing steps: expansions, updates and deletions. **Expansions** are the most frequent operation. They add new partial dialogue acts to the semantic tree. They also consult the CRF for the best surface realisation. Since CRFs are not restricted by the Markov condition, they are less constrained by local context than other models and can take non-local dependencies into account. For our application, the maximal context is 9 semantic attributes (for a surface form that uses all possible 10 attributes). While their extended context awareness can often make CRFs slow to train, they are fast at execution and therefore very applicable to the incremental scenario. For applications involving longer-spanning alternatives, such as texts or paragraphs, the context of the CRF would likely have to be constrained. **Updates** are triggered by the hypothesis updates of the dialogue manager. Whenever a new attribute comes in, it is checked against the generator's existing knowledge. If it is inconsistent with previous knowledge, an update is triggered and often followed by a **deletion**. Whenever generated output needs to be modified, old expansions and surface forms are deleted first, before new ones can be expanded in their place.

5.2 Updates and Processing Speed Results

Since fast responses are crucial in incremental systems, we measured the average time our system took for a surface realisation. The time is 100ms on a MacBook Intel Core 2.6 Duo with 8GB in

RAM. This is slightly better than other incremental systems (Skantze and Schlangen, 2009) and much faster than state-of-the-art non-incremental systems such as SPaRKy (Walker et al., 2007). In addition, we measured the number of necessary generation updates in comparison to a non-incremental setting. Since updates take effect directly on partial dialogue acts, rather than the full generated utterance, we require around 50% less updates as if generating from scratch for every changed input hypothesis. A qualitative analysis of the generated outputs showed that the quality is comparable to the non-incremental case.

6 Conclusion and Future Directions

We have presented a novel technique for surface realisation that treats generation as a sequence labelling task by combining a CRF with tree-based semantic representations. An essential property of interactive surface realisers is to keep track of the utterance context including dependencies between linguistic features to generate cohesive utterances. We have argued that CRFs are well suited for this task because they are not restricted by independence assumptions. In a human rating study, we confirmed that judges rated our output as better phrased, more natural and less repetitive than systems that just take local features into account. This also holds for a comparison with state-of-the-art rank and boost or n-gram approaches. Keeping track of the global context is also important for incremental systems since generator inputs can be incomplete or subject to modification. In a proof-of-concept study, we have argued that our approach is applicable to incremental surface realisation. This was supported by preliminary results on the speed, number of updates and quality during generation. As future work, we plan to test our model in a task-based setting using an end-to-end SDS in an incremental and non-incremental setting. This study will contain additional evaluation categories, such as the understandability or informativeness of system utterances. In addition, we may compare different sequence labelling algorithms for surface realisation (Nguyen and Guo, 2007) or segmented CRFs (Sarawagi and Cohen, 2005) and apply our method to more complex surface realisation domains such as text generation or summarisation. Finally, we would like to explore methods for unsupervised data labelling so as to facilitate portability across domains further.

Acknowledgements

The research leading to this work was funded by the EC FP7 programme FP7/2011-14 under grant agreement no. 287615 (PARLANCE).

References

Ivan Bulyko and Mari Ostendorf. 2002. Efficient integrated response generation from multiple targets using weighted finite state transducers. *Computer Speech and Language*, 16:533–550.

Hendrik Buschmeier, Timo Baumann, Benjamin Dosch, Stefan Kopp, and David Schlangen. 2012. Incremental Language Generation and Incremental Speech Synthesis. In *Proceedings of the 13th Annual SigDial Meeting on Discourse and Dialogue (SIGdial)*, Seoul, South Korea.

Nina Dethlefs and Heriberto Cuayáhuitl. 2011a. 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)*, Nancy, France.

Nina Dethlefs and Heriberto Cuayáhuitl. 2011b. Hierarchical Reinforcement Learning and Hidden Markov Models for Task-Oriented Natural Language Generation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, Portland, Oregon, USA.

Nina Dethlefs, Helen Hastie, Verena Rieser, and Oliver Lemon. 2012a. Optimising Incremental Dialogue Decisions Using Information Density for Interactive Systems. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-CoNLL)*, Jeju, South Korea.

Nina Dethlefs, Helen Hastie, Verena Rieser, and Oliver Lemon. 2012b. Optimising Incremental Generation for Spoken Dialogue Systems: Reducing the Need for Fillers. In *Proceedings of the International Conference on Natural Language Generation (INLG)*, Chicago, Illinois, USA.

Kallirroi Georgila, Nikos Fakotakis, and George Kokkinakis. 2002. Stochastic Language Modelling for Recognition and Generation in Dialogue Systems. *TAL (Traitement automatique des langues) Journal*, 43(3):129–154.

Ioannis Konstas and Mirella Lapata. 2012. Concept-to-text Generation via Discriminative Reranking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 369–378, Jeju Island, Korea.

John D. Lafferty, Andrew McCallum, and Fernando C.N. Pereira. 2001. Conditional Random

Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pages 282–289.

Wei Lu, Hwee Tou Ng, and Wee Sun Lee. 2009. Natural Language Generation with Tree Conditional Random Fields. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Singapore.

François Mairesse, Filip Jurčíček, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2010. Phrase-Based Statistical Language Generation Using Graphical Models and Active Learning. In *Proceedings of the 48th Annual Meeting of the Association of Computational Linguistics (ACL)*, Uppsala, Sweden.

Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, Genoa, Italy.

Andrew McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Crystal Nakatsu and Michael White. 2006. Learning to Say It Well: Reranking Realizations by Predicted Synthesis Quality. In *In Proceedings of the Annual Meeting of the Association for Computational Linguistics (COLING-ACL) 2006*, pages 1113–1120, Sydney, Australia.

Nam Nguyen and Yunsong Guo. 2007. Comparisons of Sequence Labeling Algorithms and Extensions. In *Proceedings of the International Conference on Machine Learning (ICML)*, Corvallis, OR, USA.

Alice Oh and Alexander Rudnicky. 2000. Stochastic Language Generation for Spoken Dialogue Systems. In *Proceedings of the ANLP/NAACL Workshop on Conversational Systems*, pages 27–32, Seattle, Washington, USA.

Matthew Purver and Masayuki Otsuka. 2003. Incremental Generation by Incremental Parsing. In *In Proceedings of the 6th UK Special-Interesting Group for Computational Linguistics (CLUK) Colloquium*.

Verena Rieser, Simon Keizer, Xingkun Liu, and Oliver Lemon. 2011. Adaptive Information Presentation for Spoken Dialogue Systems: Evaluation with Human Subjects. In *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG)*, Nancy, France.

Sunita Sarawagi and William Cohen. 2005. Semi-Markov Conditional Random Fields for Information Extraction. *Advances in Neural Information Processing*.

David Schlangen and Gabriel Skantze. 2009. A General, Abstract Model of Incremental Dialogue Processing. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, Athens, Greece.

Gabriel Skantze and Anna Hjalmarsson. 2010. Towards Incremental Speech Generation in Dialogue Systems. In *Proceedings of the 11th Annual SigDial Meeting on Discourse and Dialogue*, Tokyo, Japan.

Gabriel Skantze and David Schlangen. 2009. Incremental Dialogue Processing in a Micro-Domain. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, Athens, Greece.

Amanda Stent, Marilyn Walker, Steve Whittaker, and Preetam Maloor. 2002. User-tailored Generation for Spoken Dialogue: An Experiment. In *Proceedings of the International Conference on Spoken Language Processing*.

Matthew Stone, Christine Doran, Bonnie Webber, Tonia Bleam, and Martha Palmer. 2003. Microplanning with Communicative Intentions: The SPUD System. *Computational Intelligence*, 19:311–381.

Charles Sutton and Andrew McCallum. 2006. Introduction to Conditional Random Fields for Relational Learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press.

Sebastian Varges. 2006. Overgeneration and Ranking for Spoken Dialogue Systems. In *Proceedings of the Fourth International Natural Language Generation Conference (INLG)*, Sydney, Australia.

Marilyn Walker, Amanda Stent, François Mairesse, and Rashmi Prasad. 2007. Individual and Domain Adaptation in Sentence Planning for Dialogue. *Journal of Artificial Intelligence Research*, 30(1):413–456.