Language and Linguistics Compass 8/3 (2014): 99–115, 10.1111/lnc3.12067

Context-Sensitive Natural Language Generation: From Knowledge-Driven to Data-Driven Techniques

Nina Dethlefs*
Heriot-Watt University

Abstract

Context-sensitive Natural Language Generation is concerned with the automatic generation of system output that is in several ways adaptive to its target audience or the situational circumstances of its production. In this article, I will provide an overview of the most popular methods that have been applied to context-sensitive generation. A particular focus will be on the shift from knowledge-driven to data-driven approaches that has been witnessed in the last decade. While this shift has offered powerful new methods for large-scale adaptivity and flexible output generation, purely data-driven approaches still struggle to reach the linguistic depth of their knowledge-driven predecessors. Bridging the gap between both types of approaches is therefore an important future research direction.

1. *Introduction*

Natural Language Generation (NLG) systems across domains typically face an uncertainty with respect to the best utterance to generate in a given context. The reason is that utterances can have different effects depending on the spatial and temporal environment, addressee and interaction history that characterise the *context* in which they occur. Stalnaker (1998) defines context as a set of knowledge, which holds at the time of the discourse and constrains the semantic content of each generated utterance. Knowledge is shared between all discourse participants who act upon and modify it continuously as new discourse is created. In the case of NLG, context can thus be seen as a dynamic notion, which influences both the generation of utterances and their interpretation by the user. Under this view, any utterance made in the discourse will update the context by either adding or eliminating knowledge from it.

In this article, we will distinguish two aspects of context-sensitive NLG: *audience design* and *situation design*. *Audience design* refers to generation that adapts to individual users or groups of users. Adaptation can be towards the user's prior knowledge of a domain or the task at hand, such as *expert* vs *novice*, or it can address individual preferences such as the *cheapest* flight to a destination vs the *most comfortable* or the *shortest* flight. Challenges in audience design are often to identify such preferences, classify new users correctly and determine what adaptation exactly may be appropriate for a particular user group. *Situation design*, in contrast, addresses adaptation to spatial and temporal properties. It can be defined as generation that is explicitly adaptive to an enriched physical context, including features of a (real or virtual) environment, such as spatial objects or users. The context in this setting is typically not static but undergoes dynamic changes due to the environment changing over time or users manipulating it.

The GIVE Challenge (Byron et al. 2009; Koller et al. 2010) is a good example of a domain that requires both types of adaptation. Here, two participants, one instruction giver and one instruction follower, engage in a 'treasure hunt' through a set of virtual worlds. The task can be won by finding and unlocking a safe and obtaining a trophy from it. To solve the task, the instruction giver has to guide the instruction follower in navigating through a world and pressing a particular sequence of buttons. The sequence of buttons corresponds to a code that will, if pressed in the correct order, unlock the safe and release the trophy. There are also a number of distractor buttons present, though, which either have no effect or trigger an alarm. In the original GIVE task, the role of the instruction giver is taken by an NLG system, which, among others, needs to generate referring expressions that are sensitive to different configurations of the environments. Importantly, each successful referring expression should have an effect on the generation situation. For example, if the system suggests a user to *press the green button to open the door*, and the user complies, the door should now be open in the updated environment. A challenge is often to keep track of dynamic environment updates and determine what adaptation is appropriate in a particular situation.

Figure 1 shows a potential application for situation design within the GIVE scenario. We see different situations in which the user should press a single button (circled), called the referent. All other buttons are distractors, and pressing one of them would have an unknown negative effect, in the worst case the game is lost instantly. The NLG system therefore needs to generate an unambiguous reference and faces a number of choices. For the first scene, for example, it can say *Push the button in the middle on the very right*, using spatial references to locate the button. In the second scene, *Push the red button* would suffice, but *Push the left button* and *Push the button beside the plant or the button left of the green* are also acceptable, differing in their use of colour and positioning with respect to distractors, objects or spatial references. In the third scene, none of the previous strategies are likely to be successful. Something more descriptive may therefore be needed such as *Push the second top-most button left of the red*. However, generation may also take the user's prior knowledge of the task into account and generate, for example, *Push the same as before*, or *Open the door*, assuming that this is the button's function and the user is likely to know this. The latter utterances are examples of audience design in the GIVE task.

Different methods have been suggested for context-sensitive NLG, which we will review in turn. All of these methods need some way to address the core tasks that NLG has conventionally been divided into: *content determination*, *sentence planning* and *surface realisation* (see Reiter 1994; Reiter and Dale 2000) for details). Content determination is responsible for constructing a semantic representation from the initial NLG system input. It determines *what to say*, given the current communicative intent. The next stage, sentence planning, then maps the semantics onto a set of sentences and clauses, applying aggregation and choosing grammatical structure. Referring expression generation is also part of sentence planning. Finally, surface realisation finds a linguistic realisation for each semantic input.

The article is organised as follows. In Sections 2 and 3, we will begin with rule-based and planning accounts, which have in many ways laid the foundation to the field of context-sensitive NLG. These approaches will be referred to as *knowledge-driven*. Subsequently, in Section 4, we will compare a range of methods that make use of a machine learning

Fig 1. Scene from a situated context-sensitive NLG task. The NLG system has to generate an unambiguous reference to the (circled) referent button. It has several strategies for doing so, but the best strategy will depend on the physical context, the distractors and the user's prior knowledge.

component, such as supervised learning, unsupervised learning or reinforcement learning. We refer to these approaches as *data-driven*. The comparison will place a particular focus on the influence that the general shift from knowledge-driven to data-driven methods experienced in Natural Language Processing in the last decades has had on research in context-sensitive NLG. On the one hand, this shift has offered a new set of tools and methods that have allowed us to address issues of robustness and adaptivity that former knowledge-driven systems have been restricted by. On the other hand, the takeover of statistical methods has opened up new challenges that become manifest in data-driven systems still struggling to match the level of linguistic understanding of their earlier predecessors. Finally in Section 5, we conclude with a discussion of the remaining hurdles for fully flexible and extensible context-sensitive NLG and how future research can move towards overcoming them.

2. Context-sensitive NLG as Explicit Choice: Rule-based Approaches

Rule-based context-sensitive NLG can be seen as a process of explicit choice in which the generation system faces a sequence of context-dependent choices that ultimately lead deterministically to a single output. For example, given an end user, a situation and a knowledge base of facts, which of those facts should be included in a semantic form for presentation to the user? This scenario is an example of content determination in context-sensitive NLG, which was first studied in foundational work by McKeown (1985).

Algorithm 1 Full Brevity Algorithm for Generating Referring Expressions

Let L be the set of properties to be realised in a description; let P be the set of properties known to be true of our intended referent r (we assume that P is non-empty); and let C be the set of distractors (the contrast set); The initial conditions are thus as follows:

$C = \{\langle\text{all distractors}\rangle\}$
$P = \{\langle\text{all properties true of } r\rangle\}$
$L = \{\}$
In order to describe the intended referent r with respect to the contrast set C, do the following:

1. Check Success:
if $|C| = 0$ **then** return L as a distinguishing description
else if $P = 0$ **then** fail
else goto Step 2.
end if $P = \emptyset$ fail

2. Choose Property:
for each $p_i \in P$ **do**:
    $C_i \leftarrow C \cap \{x \mid p_i(\mathrm{x})\}$
    Chosen property is p_j, where C_j is the smallest set.
    goto Step 3.
end for

3. Extend Description (wrt the chosen p_j):
$L \leftarrow L \cup \{p_j\}$
$C \leftarrow C_j$
$P \leftarrow P - \{p_j\}$
goto Step 1.

For an alternative example, consider Algorithm 1, which specifies a set of rules for generating uniquely identifying referring expressions and could, among many others, be applied to the scene in Figure 1. The Full Brevity Algorithm (Dale 1989) aims for minimally distinguishing descriptions of a referent and therefore always chooses the property that is most discriminating among the available ones. These properties can include the colour of a referent, position, shape, etc. depending on the generation domain. Once a uniquely identifying description has been found, no further detail is included. This algorithm represents one of the many ways to solve referring expression generation. For a recent overview, see Viethen (2011).

The question of how to express a content once a particular semantic representation has been found is addressed in early work by Bateman and Paris (1989). They generate descriptions of digital circuits that are specifically tailored towards the user's level of domain expertise, such as expert or novice. This is achieved by varying the level of technical detail and the syntactic formality. Three examples, for different target user groups, are shown in Figure 2. This scenario is an example of audience design. Related work includes those of Paris (1993) and Bateman and Teich (1995), who generate user-tailored technical descriptions and genre-specific biographies, respectively.

Many of the early approaches towards rule-based context-sensitive NLG are couched within various computational grammar formalism, such as *Unification Grammars* (Elhadad 1993; Kay 1985), *Tree Adjoining Grammar* (Joshi 1987; Stone and Doran 1996) or *Systemic Functional Grammar* (Bateman 1997). Other approaches include *Meaning-Text Theory* (Mel'cuk 1988; Lavoie and Rambow 1997), *Unification Categorial Grammar* (Calder et al. 1989) or classification (Reiter and Mellish 1992). Ward (1994) gives a comprehensive overview of the generation as choice paradigm.

More recent work on audience design using rules in context-sensitive NLG includes Androutsopoulos et al.'s (2007) M-PIRO system. M-PIRO is an NLG system that generates personalised descriptions of museum artefacts in multiple languages, adapting both its semantic and lexical-syntactic features. M-PIRO may, for example, present less concrete facts to a child than an adult and choose less complex sentence structures.

Another system that aims to adapt to an audience at several levels is White et al.'s (2010) FLIGHTS system. It presents flight-related information in a way that is particularly tailored towards individual users in terms of content determination, selection of referring expressions,

---

Text generated for expert users (system developers)

The system is faulty, if there exists a 0 in the set of the output terminals of the system such that the expected value of the signal part of 0 does not equal the actual value of the signal part of 0 and for all 1 in the set of the input terminals of the system, the expected value of the signal part of 1 equals the actual value of the signal part of 1.

Text generated for non-expert users (who want to follow the system's reasoning

The system is faulty, if all of the expected values of its input terminals equal their actual values and the expected value of one of its output terminals does not equal its actual value.

Text generated for Naive Users (who are just interested in the status of the system)

The system is faulty, if the inputs are fine and the output is wrong.

Fig 2. Example texts from Bateman and Paris (1989) tailored towards an expert audience of system developers, interested non-expert users, who want to understand the status of the system, and a non-expert audience.

information structure and realisation units for speech synthesis. All of these decisions are made in an integrated fashion so as to ideally emphasise the user preferences and arising trade-offs. For a user preferring to travel on the cheapest flight, for example, the system will present the cheapest flight using a prosodic realisation that appropriately puts the price attribute in focus, such as 'the *cheapest* flight', where cursive fonts indicate prosodic emphasis. White et al. present a human evaluation showing that users significantly prefer the emphatic prosody of the system over its baselines.

In contrast to the above approaches, Denis (2010) presents an application of situation design. His approach to referring expression generation in the GIVE task works by systematically eliminating distractor buttons until a unique reference to the target referent is possible. This is based on the observation that language use is not only determined by a context but also updates it. For example, generating the referring expression *the blue button* may still leave us with several candidates, but we have definitely excluded the set of red ones. The next utterance can then build up on this newly created context, e.g. saying *the left one*, or *yes, this one!* when the user hovers their mouse over the intended referent. Such a strategy may be useful in the right-most situation in Figure 1 and achieves a task success of over 90% reported by Denis (2010). This approach is quite different from Algorithm 1. While Dale's (1989) approach generates referring expressions that are discriminated within the current context, it does not take their subsequent impact on the (updated) context into account.

To integrate both audience design and situation design in a single system, Dethlefs et al. (2011) present an outdoor route instruction generation system. Route instructions are sensitive to the user's familiarity with the navigation area as well as to salient geographical properties, such as prominent streets or landmarks. They present results indicating a significant human preference for adaptive over non-adaptive route instructions.

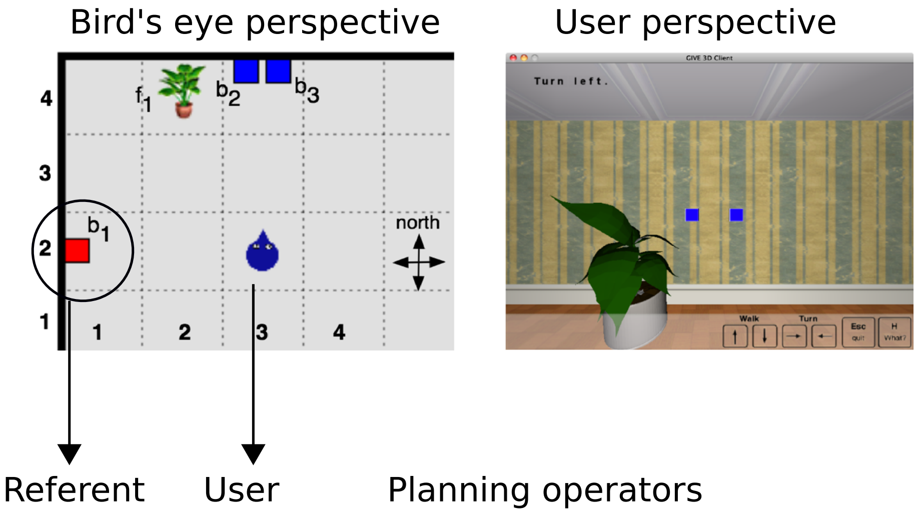
The most important advantage of deterministic knowledge-driven approaches is that they give complete control to the system designer in the specification of detailed linguistically or psychologically informed knowledge. Rule-based systems therefore often achieve high levels of sophistication within their domain.

Disadvantages, on the other hand, include the time and cost involved in system development and maintenance. Developing an NLG system from scratch requires a detailed analysis of the target domain, identification of generation features and meticulous rule specification. In addition, the quality of a rule-based NLG system can depend to a large extent on the skills and experience of the individual system designer – maybe more so than in the case of statistical methods. Finally, while rule-based systems achieve high performance within their target domain, they can be brittle when faced with unseen system input.

3. Context-sensitive NLG as Problem Solving: Planning Approaches

Early approaches to planning have typically defined actions with respect to a set of preconditions and effects that capture linguistic or contextual knowledge and can be used to meet a set of pre-specified constraints (McDonald 1976; Patten 1988; Appelt 1992; Cohen and Perrault 1986; Heeman and Hirst 1995). Several approaches have also been investigated using hierarchical planning, such as those of Moore and Paris (1994), Hovy (1991) or Appelt (1992). While these approaches do not explicitly address context-sensitive NLG, they can be said to lay the foundations for modern planning approaches.

As an example of context-sensitive planning, consider Figure 3. It contains a spatial scene (from a bird's eye perspective and from the user's perspective) in which two buttons are visible to the user. The referent button b_1, however, is not visible. The system could now formulate a complex instruction such as *Turn 90 degrees to the left and press the red button.*



turn left(u, x, o_1, o_2):
Precond: $\mathrm{subst}(S, u)$, $\mathrm{ref}(u, x)$, $\text{player-ori}(o_1)$ $\text{next-ori-left}(o_1, o_2)$, ...
Effect: $\neg\mathrm{subst}(S, u)$, $\neg\text{player-ori}(o_1)$, $\text{player-ori}(o_2)$ $\text{to-do}(\mathrm{turnleft})$, ...

push$(u, u_1, u_n, x, x_1, p, o)$:
Precond: $\mathrm{subst}(S, u)$, $\mathrm{ref}(u, x)$, $\text{player-pos}(p)$, $\text{player-ori}(o)$, $\mathrm{visible}(p, o, x_1)$, ...
Effect: $\neg\mathrm{subst}(S, u)$, $\mathrm{subst}(NP, u_1)$, $\mathrm{ref}(u_1, x_1)$, $\forall y.\ (y \neq x_1 \wedge \mathrm{visible}(p, o, y) \rightarrow \mathrm{distractor}(u_1, y))$, $\text{to-do}(\mathrm{push}(x_1))$, $\mathrm{canadjoin}(S, u)$, ...

and(u, u_1, u_n, e_1, e_2):
Precond: $\mathrm{canadjoin}(S, u)$, $\mathrm{ref}(u, e_1)$, ...
Effect: $\mathrm{subst}(S, u_1)$, $\mathrm{ref}(u_1, e_2)$, ...

Fig 3. Planning operators specifying preconditions and effects for linguistic and non-linguistic actions in the GIVE world. The example situation is shown from a bird's eye perspective in the upper left-hand corner, where b_1 is the intended referent. The same scene is shown from the user's perspective in the upper right-hand corner. Adapted from Garoufi and Koller (2010).

Alternatively, it could generate a set of simpler instructions, easing the user's cognitive load. Garoufi and Koller (2010) suggest to follow the latter approach. Their planning operators (shown in the bottom of Figure 3) describe the non-linguistic conditions that are required in the current situation to produce an utterance. In the example, a precondition for a referring expression is that the user can see the referent. One of the effects of telling the user to *turn left* is to fulfil this condition: as soon as the user turns left, b_1 will become visible. The system will therefore generate two instructions: *Turn left* and *Push the button*. This clearly demonstrates the benefits of making systematic use of the non-linguistic context of generation by not only using the context to inform language generation but also manipulating the non-linguistic context by means of language generation.

Other work on context-sensitive planning and NLG includes Piwek and van Deemter (2007), who use constraint satisfaction to generate scripted dialogues, and Golland et al. (2010) who treat generation as a game-theoretic model in which a set of semantic and pragmatic constraints need to be satisfied. See Koller and Petrick (2011) for a recent overview of planning approaches for NLG.

Among the most important advantages of planning-based approaches is that they typically meet their specified goals reliably and often have several ways of doing so, in case one of them fails. This makes them robust and flexible in complex domains and has helped them to solve considerably more complex problems than is to date possible with data-driven techniques. As an example of the linguistically and rhetorically rich knowledge involved in some planning approaches, consider Figure 4. It is adapted from Moore and Paris (1994) and shows an example dialogue between a user and NLG system generated by the PEA system (Neches et al. 1985). The system generates explanations of programmes based on detailed linguistic features and rhetorical relations. Such knowledge provides the system with a deep understanding of the conversational context and the contributions of its own utterances. It is therefore able to monitor and reason about the effects that its own utterances will have on the user and the conversational context. In this way, the NLG system can make inferences about previous utterances or react to follow-up questions from the user, provide clarifications or elaborate on previous utterances. In Figure 4, the user should now, for example, be able to ask a follow-up question such as *What is an accessor function?* or *In what way does this improve the readability of my programme?* The system should then be able to identify the relevant subpart of the utterance that requires clarification.

Similar reasoning skills for NLG can be found in the Witas system (Lemon et al. 2002). It addresses NLG within the setting of an autonomous helicopter robot, which manages collaborative tasks and conversations with a human user. In particular, the system deals with several communication streams at the same time as new information comes in through its activity sensors, which needs to be communicated to the human user. Similar to the previous example, the human operator can make specific queries about the robot's actions and plans, such as its heading direction, next task, etc., which then become the focus of the conversation. To reply to these queries, it will need rich contextual information and awareness of the reasoning behind its actions, especially since communication goals often arise out of the context of the conversation, such as a previous or another participant's utterance. An example dialogue is shown in Figure 5.

Knowledge-aware reasoning as in the above examples can often be found in planning approaches. Since the NLG system needs to reason about its next action, taken the desired effects into account, the ability to verbalise plan steps is inherent in the approach if enabled by the system designer. It can still sometimes pose a challenge for data-driven techniques, though, that often work based on probabilistic mappings between inputs and outputs.

Disadvantages of planning approaches are partially shared with rule-based approaches. Designing a new planning-based NLG system can require substantial amounts of time invested in domain analysis, and constraint specification and its success will critically depend on the skills of the system designer. While generalisation to new situations is possible within the defined domain, constraints may fail to generalise to unseen system inputs and lead to unpredictable system output.

System:	What characteristics of the programme would you like to enhance?
User:	Readability and maintainability.
System:	You should replace `(SETQ X 1)`with `(SETF X 1)`. `SETQ` can only be used to assign a value to a simple-variable. In contrast, `SETF` can be used to assign a value to any generalised-variable. A generalised-variable is a storage location that can be named by any accessor function.

Fig 4. Example dialogue between a user and system in which the system generates explanations of programmes. The user is able to ask follow-up questions, request elaborations or clarifications. Adapted from Moore and Paris (1994).

User:	Our job is to look for a red car.
System:	OK, I am looking for one.
User:	Fly here, please [+ click on map]
System:	Okay. I will fly towards waypoint one. Now taking off and flying there.
User:	Stop that. Go to the tower instead.
System:	I have cancelling flying to waypoint one. I will fly to the tower.
User:	What are you doing?
System:	I am searching for a red car and flying to the tower.

Fig 5. Example dialogue between an autonomous helicopter robot and a human operator. The robot carries out tasks given by the operator and can communicate plans and intentions. Adapted from Lemon et al. (2002).

4. Context-sensitive NLG as Optimisation: Machine Learning Approaches

Most recent approaches towards context-sensitive NLG contain some form of machine learning, such as supervised learning, unsupervised learning or reinforcement learning. For work on active learning for general NLG, see Mairesse et al. (2010). As in planning, these approaches typically start from an overall goal to achieve, specified by the system designer, and then use machine learning to automatically discover the best strategy to achieve it.

The general idea of data-driven generation was first introduced as an over-generation and ranking task in seminal work by Langkilde and Knight (1998), followed by work of Oh and Rudnicky (2000), Bangalore and Rambow (2000) or Ratnaparkhi (2000). The idea of over-generation and ranking approaches is that a system generates a large number of candidate utterances in a first step, which are then ranked according to a pre-specified measure of performance in a second step. Typically, ranking is performed according to the frequency of occurrence of utterances in a corpus. The approaches above find the most likely surface realisation for a semantic representation based on n-gram models. An n-gram model predicts word w_i as

$$P(w_i) = P\big(w_i | w_{i-1}, w_{i-2}, \dots w_{i-(n-1)}, a\big), \tag{1}$$

where w denotes the word sequence of the n-gram and a represents the attribute type, such as referring expression. If there is just one attribute, a can be omitted. While none of these approaches focused on context-sensitive NLG, they can be said to mark the beginning of data-driven, or trainable, NLG in general. For further reading on the field of machine learning, Mitchell (1997) and Bishop (2006) are good references.

4.1. SUPERVISED LEARNING

The general idea behind supervised learning is to learn a mapping function between a set of observations, the inputs, and a set of explanatory variables, the outputs. To induce a supervised learner from data, one usually starts from a set of labelled training data. The data consist of input-output pairs of the form $\{(x_1,y_1), (x_2,y_2) \dots (x_N,y_N)\}$, and the task of the learner is to find a mapping function from input X to output Y, where X is the set of possible inputs and Y is the set of possible outputs. Such a function can be used by a classifier according to

$$y^* = \arg\max_y f(x, y), \tag{2}$$

where y^* represents the highest scored label and the f function can be found by supervised learners including Naive Bayes methods, decision tree learners, rule-based learners, instance-based learners and many more.

As an example of a supervised decision maker, consider the decision tree shown in Figure 6. This tree was trained from the GIVE corpus (Gargett et al. 2010), a corpus derived from a set of human-human dialogues collected in the GIVE world. Its classification task is to predict whether or not to include a referent's colour in a referring expression. To do this, a typical input to the tree is a feature vector, maybe of the form *discriminatingColour* = *true*, *repair* = *false*. A possible output is then *useColour* = *true*. The tree indicates a clear human preference for using colour, even when it is not uniquely identifying. According to the data, humans always mention a referent's colour, unless the colour is not discriminating and the current utterance is a repair (such as rephrasal) of a previous utterance. This again is a different strategy than that shown in Algorithm 1.

An application of decision trees in a task similar to GIVE, but involving spoken language, is presented by Stoia et al. (2006). The authors design a classification-based algorithm for noun phrase generation based on physical properties such as the user's viewing angle, distance from target, visibility, etc. They show that humans judged their generation output as equivalent or better than the original human noun phrases that the classifier was trained on in 62.6% of cases.

For an application of corpus-based language modelling in multimodal systems, see Foster and Oberlander (2006). They generate facial displays for a talking head, which are sensitive to the utterance context in terms of its words, its pitch-accent specification and its domain context (bathroom tile design). The authors show that users prefer context-sensitive generation models that at the same time introduce variation (probably to make the facial displays more interesting).

Walker et al. (2007) apply a boosting technique to the task of individualised surface realisation in the SPaRKy system. Their approach works by generating a set of possible outputs in a first stage, which are then ranked according to predicted user preferences in a second stage. Results showed that while humans rate SPaRKy sentences significantly better than random sentences, they rate them on average 10% worse than human-generated sentences.

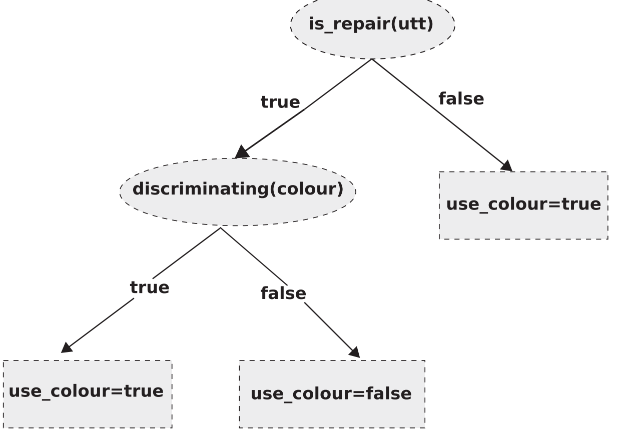


Fig 6. Decision tree trained on the GIVE corpus for the question of when to include a referent's colour in a referring expression. Here, ellipses represent decision points in the tree (random variables), arrows represent their values and rectangles represent classification output. Training was done using the Weka toolkit (Witten and Frank 2005).

Trainable approaches to generation such as supervised learning, example-based learning (DeVault et al. 2008; Stent et al. 2008) or n-gram models perform best when trained from a large number of well-balanced and representative training examples. This can make them faster to develop than knowledge-driven approaches (given that training data are available) and can replace human intuition in areas where human analysis is costly or the data are diverse and difficult to analyse. It also removes the development variable of the system designer's particular skills. Supervised learning-based NLG systems have been shown to reach comparable performance to knowledge-driven approaches for certain domains and are typically more robust to unseen system input.

A problem in supervised learning is that the system quality depends entirely on the underlying training data. If trained from a small or biased corpus, system behaviour can become unpredictable in unseen situations (Levin et al. 2000). Therefore, whenever no suitable data are available for the target domain, data collection and annotation can become a serious overhead comparable to the effort required in designing a rule- or planning-based system.

4.2. UNSUPERVISED LEARNING

In contrast to supervised learning, unsupervised learning assumes that all explanatory variables are latent and need to be identified from a corpus of unlabelled examples. Among the most important advantages of unsupervised learning methods is therefore that they can be trained directly from unlabelled input, making them cheap to train and port across domains.

As an example of this, Roth and Frank (2010) present a largely unsupervised learning approach to the generation of route instructions for outdoor environments, which is again an application of situation design. They use the Expectation-Maximisation (EM) algorithm to align geographical route representations with corresponding linguistic realisations that were taken from an annotated corpus of human data and show that their model performs better than a random baseline.

As an example of unsupervised learning for audience design, Demberg and Moore (2006) present an algorithm that combines traditional user modelling with clustering for information presentation in the flight domain. They use a cluster-based tree structure to guide the presentation of information for particular users based on the user's interests and generally relevant information. They present results showing a significant increase in user satisfaction over a baseline.

Alternative approaches have also used a corpus of textual descriptions, which were unlabelled, but aligned with non-linguistic contexts to induce NLG systems. Chen et al. (2010) have learnt an NLG system from a corpus of textual descriptions that were aligned with actions in a RoboCup soccer game. While the mapping from descriptions to action sequences was given, the input was ambiguous in that not all actions were included in the descriptions.

Benotti and Denis (2011) learn an NLG system from unlabelled human instructions in the GIVE task, which were aligned with situations in the virtual world. For example, if an instruction *Go through the door and push the button* elicits a user reaction consisting of just this action sequence, the NLG system can learn the specific sequence of instructions that will yield a certain effect.

A distinct advantage of unsupervised learning approaches is that they offer the possibility to train an NLG system from unlabelled data, which significantly reduces development costs and facilitates portability across domains.

A problem is that the saved development costs in unsupervised learning have so far always come at the cost of semantic sophistication. In fact, all unsupervised NLG systems discussed above use hybrid techniques, i.e. they use unsupervised learning in combination with another, semantically richer, technique. It has not yet been possible to induce an NLG system from unlabelled data and achieve system behaviour that adapts to different users or contexts in a way similar to knowledge-driven or supervised learning approaches.

4.3. REINFORCEMENT LEARNING

In contrast to the above learning methods, Reinforcement Learning (RL) agents do not learn from examples but from a trial and error search. To do this, four components typically need to be specified: a *state representation*, an *action set*, a *transition function* and a *reward function*. The state representation usually consists of a discretised feature set representing the generation context. It could, for example, include the present number of distractors and landmarks, whether the referent's colour is discriminating, etc. The action set contains all generation actions available to the agent, such as mention the referent's colour, mention a distractor, etc. The transition function updates the state representation after each action to represent the action's effect on the context, and the reward function, finally, allows the agents to evaluate its actions. For the GIVE task, one could assign a positive reward for generating a uniquely identifying reference and a negative or other numerical reward otherwise. The agent would then try different action sequences in different contexts and discover the most rewarding generation strategy in the long term according to

$$\pi^*(s) = \arg\max_a Q^*(s,a), \tag{3}$$

where π^* denotes the optimal policy of mapping action a to state s and Q^* represents the optimal mapping from one state to an action. See Sutton and Barto (1998) for a detailed introduction to RL.

Janarthanam and Lemon (2010) present an interesting application of RL to audience design in the context of a spoken dialogue system, which helps users set up their home broadband connection. Adaptation occurs at the level of lexical choice in referring expressions. For example, an expert user may know what the term *broadband filter* refers to, but for a non-expert, the term *small white box* may be more helpful. Note that this example is surprisingly similar to the research of Bateman and Paris (1989) two decades earlier. Two features mark the difference. First, while Bateman and Paris specified a generation strategy based on a human analysis of the target domain, Janarthanam and Lemon use learning from a simulated user to discover the best strategy automatically. Second, while Bateman and Paris made the assumption that the generation context was static, that is that the user's technical knowledge did not change at least for a single generation episode, Janarthanam and Lemon assume a dynamic context in which the user is able to learn new technical jargon during the interaction.

Reinforcement learning is in several ways related to planning approaches to NLG and can be seen as an approach to *statistical planning*. While both planning and reinforcement learning approaches can find more than one way to achieve a generation goal, an advantage of RL approaches is that they can test and evaluate different alternatives during training and optimise their performance given their observed effects and rewards. This is particularly helpful when optimising action sequences in the long term rather than single generation decisions.

Of course, RL-based approaches have drawbacks. One of the most important disadvantages is the *curse of dimensionality*. This refers to the fact that an RL agent's state space grows exponentially with the number of state variables taken into account. This causes learning to be slow and makes it difficult to discover optimal policies for complex domains with large state spaces. Possible remedies include function approximation techniques (Mitchell 1997;

Bishop 2006), or the use of a divide-and-conquer approach. The latter is a form of hierarchical RL in which a generation task is decomposed into a number of subtasks for which optimal policies can be found more easily. This technique was first suggested by Cuayáhuitl (2009) for dialogue management and by Dethlefs and Cuayáhuitl (2010, 2011) for NLG.

Figure 7 contrasts flat and hierarchical RL for referring expression generation in GIVE. While in the flat setting, all decisions are made by one agent, the hierarchical setting is decomposed into four subtasks: making decisions about the referent, decisions about distractors, about landmarks and about spatial relations. Importantly, the state-action space of the hierarchical agent corresponds to only 2% of the original flat state-action space. This dramatic reduction can speed up learning and allow learning for complex domains.

A further disadvantage of RL approaches is the effort involved in specifying a simulated environment and a reward function for learning. This is comparable to the work involved in specifying rules or constraints in knowledge-driven approaches or the effort in data collection and annotation of supervised approaches. This is an active research field, and first approaches have been presented to induce both automatically from data (Walker et al. 1997; Rieser et al. 2010; Dethlefs and Cuayáhuitl 2011).

(a)

Feature	Values	Description
f_0	true,false	Is the referent's colour discriminating?
f_1	none,one,few,many	How many distractors are present?
f_2	none,one,few,many	How many landmarks are present?
f_3	true,false	Is the referent's spatial position discriminating?
f_4	null,yes,no	Has colour been mentioned?
f_5	null,yes,no	Has a distractor been mentioned?
f_6	null,yes,no	Has a landmark been mentioned?
f_7	null,yes,no	Has a spatial position been mentioned?

Action	Description
a_0	Mention referent colour
a_1	Do not mention referent colour
a_2	Include distractor
a_3	Do not include distractor
a_4	Include landmark
a_5	Do not include landmark
a_6	Include spatial position
a_7	Do not include spatial position

(b)

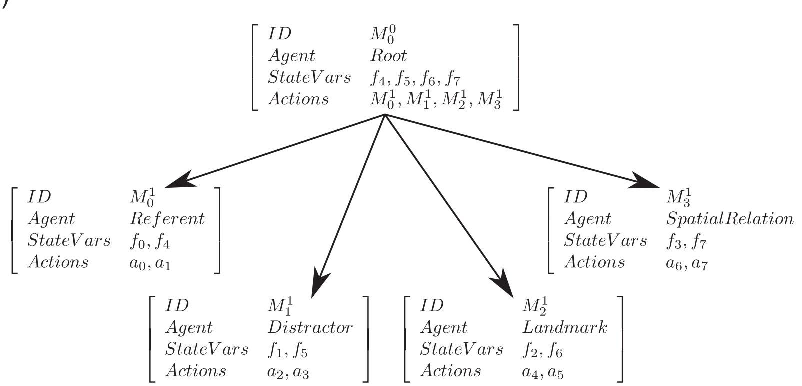


Fig 7. A comparison of a flat (a) and hierarchical (b) state-action space for an RL agent for referring expression generation. While the flat agent has 41,472 state-actions, the hierarchical agent has only 420, dramatically reducing the search space for the learning agent.

5. Conclusion

Context-sensitive NLG systems that adapt their output to different users or situations have potential application in a variety of domains. We have seen generation in applications such as technical descriptions or user manuals, instruction generation in dynamically changing indoor and outdoor environments, and generation of utterance contributions in the travel domain and other dialogues. Further possible applications include the generation of weather reports (Belz and Reiter 2006), medical information (Mahamood and Reiter 2012), tutoring dialogues (Jordan et al. 2012), multimodal generation (Beun and Cremers 1998; van der Sluis and Krahmer 2007), incremental generation (Skantze and Hjalmarsson 2010; Dethlefs et al. 2012; Buschmeier et al. 2012) and many more.

All of the above can be treated as instances of context-sensitive NLG, and we have seen a multitude of paradigms to address such applications. Knowledge-driven systems have typically been designed through meticulous domain analysis, feature identification and knowledge specification and have reached high performance within their domain. Some domains can be difficult to analyse because the data (if available) are diverse or there are multiple strategies to achieve a goal. In such cases, machine learning methods can help in the analysis and often find a good NLG strategy automatically – from labelled examples in supervised learning and from evaluative feedback in reinforcement learning. Reinforcement learning is further suitable for sequential decision making problems, for example, when trying to not just optimise a single generator output but try a sequence of them in order to find the best strategy in the long term. The quality of both supervised and reinforcement learning-based systems currently depends on the quality of the available domain data. In supervised learning, the learnt system will only be as good as the labelled examples. In reinforcement learning, all system behaviour is determined by the simulated environment and the reward function. While the collection and annotation of a well-balanced and representative data set can be daunting, it typically makes data-driven approaches more robust to unseen system inputs than knowledge-driven approaches such as rule-based or planning systems. The cheapest method in terms of system design is probably unsupervised learning techniques, which can be trained from unlabelled examples. However, the speedy development usually comes at the cost of semantic sophistication. Domingos (2012) gives an overview of machine learning including principles for choosing among different methods, common principles and pitfalls to avoid.

In NLG research, the best method to use often depends on the particular task at hand. Data-driven techniques offer a number of new tools and distinct advantages over knowledge-driven approaches. On the other hand, as illustrated on some of the knowledge-intensive planning examples above, they have not yet been able to reach a comparable level of domain understanding and sophistication to earlier semantically rich approaches.

We can identify six main obstacles, here called *challenges*, that affect context-sensitive NLG in one way or another and that future research will need to overcome.

1. *The Human-Labour Challenge* refers to the problem of dealing with the large amount of human effort involved in system design. Specifically, it can refer to rule or constraint specification to semantic annotation for supervised learning or the design of a simulated environment for reinforcement learning.
2. *The Data Challenge* denotes really two separate, but related, problems. The first is that for any new domain a researcher wishes to address, they will typically require domain-specific data, which can be difficult and costly to obtain. Secondly, we have a large amount of unstructured data available on the world-wide web or social media. Unfortunately, we do not yet have data mining methods that are powerful enough to deal with these data fully automatically and on a large scale.

3. *The Feature Selection Challenge* is mostly relevant to machine learning applications, and also to knowledge-driven approaches. For the former, it refers to the task of selecting exactly those features that will help the agent find useful patterns in the data without the risk of over-fitting. For the latter approaches, it refers to the challenge in data analysis that a human designer faces when deciding which features to include in an algorithm and which not.
4. *The Scalability Challenge* is again related to the current absence of powerful methods in the field that can deal with large amounts of data or knowledge in a scalable way. Several approaches exist that aim to tackle complex and ambitious goals (Janarthanam and Lemon 2010; Rieser et al. 2010; Pietquin et al. 2011), but they are currently confined to small-scale applications.
5. *The Generalisability Challenge* refers to the fact that systems can almost never act in any domain other than the one they were explicitly designed or trained for because their generalisation abilities are very restricted or non-existent. It is likely that in order to build more advanced applications and extend our research to real-world problems, we will need to investigate methods that endow systems with a particular skill set of domain-independent strategies.
6. *The Evaluation Challenge*, finally, addresses the fact that in contrast to other areas, such as parsing or automatic speech recognition, where gold standards for evaluation are clearly defined, the quality of NLG output is often a matter of subjective assessment. Several objective metrics have been applied to NLG, such as accuracy, similarity with human data, efficiency or task success, but they are of little help when evaluating concepts such as naturalness, phrasing or coherence. Evaluation challenges have begun to establish standards and grounds for comparison in recent years, but more work in this direction is needed.

Future research can take many directions but will in the long term need to address the problems arising from the challenges above. If we can find methods that help us deal with the large amounts of data available and make sense of their semantic and pragmatic properties in a scalable way, this may also help us to bring the deep semantic understanding of knowledge-driven approaches and the flexibility and robustness of data-driven approaches closer together.

Short Biography

Nina Dethlefs is currently a Research Associate at the Interaction Lab in the School for Mathematical and Computer Sciences at Heriot-Watt University in Edinburgh. Her research interests focus on Natural Language Generation (NLG) in the context of interactive systems including incremental and situated Natural Language Generation. She is interested in Machine Learning methods such as reinforcement learning as well as graphical models and various hybrid stochastic approaches. Previous to joining Heriot-Watt University, Nina was a PhD student and Research Assistant in the Computational Linguistics and Spatial Cognition groups at the University of Bremen, Germany. Her PhD research focused on a joint optimisation approach towards decision making in a situated Natural Language Generation system using hierarchical reinforcement learning. From 2004 to 2008, Nina did her undergraduate studies at the Universities of Bremen and Edinburgh. Between 2009 and 2010, she worked on collaborative projects on generating adaptive spoken route instructions with the University of Melbourne, Australia, and on reference generation in situated dialogue with the Centre for Language Technology at Macquarie University in Sydney.

Acknowledgements

Many thanks to Heriberto Cuayáhuitl and the anonymous reviewers for very helpful comments on an earlier draft of this paper. The work was partially supported by the EC FP7 project PARLANCE under grant agreement number 287615.

Notes

* Correspondence address: Nina Dethlefs, Heriot-Watt University, Mathematical and Computer Sciences (MACS), Riccarton, Edinburgh EH14 4AS, Scotland, UK. E-mail: n.s.dethlefs@hw.ac.uk

Works Cited

Androutsopoulos, I., J. Oberlander, and V. Karkaletsis. 2007. Source authoring for multilingual generation of personalised object descriptions. Natural Language Engineering, 191–233.

Appelt, D. 1992. Planning English Sentences. New York, NY, USA: Cambridge University Press.

Bangalore, S., and O. Rambow. 2000. Exploiting a probabilistic hierarchical model for generation. Proceedings of the 18th Conference on Computational Linguistics, 42–48.

Bateman, J. 1997. Enabling technology for multilingual natural language generation: the KPML development environment. Natural Language Engineering 3(1) 15–55.

——, and C. Paris. 1989. Phrasing a text in terms the user can understand. Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI), Detroit, Michigan, USA, 1511–1517.

——, and E. Teich. 1995. Selective information presentation in an integrated publication system: an application of genre-driven text generation. Information Processing and Management: An International Journal 31(5) 753–767.

Belz, A., and E. Reiter. 2006. Comparing automatic and human evaluations of NLG systems. Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Trento, Italy.

Benotti, L., and A. Denis. 2011. Giving instructions in virtual environments by corpus-based selection. Proceedings of the 12th Annual Meeting on Discourse and Dialogue (SIGdial), Portland, Oregon.

Beun, R.-J., and A. Cremers. 1998. Object reference in a shared domain of conversation. Pragmatics and Cognition 6(1) 111–142.

Bishop, C. 2006. Pattern recognition and machine learning. New York: Springer.

Buschmeier, H., T. Baumann, B. Dosch, S. Kopp, and D. Schlangen. 2012. Incremental language generation and incremental speech synthesis. Proceedings of the 13th Annual SIGdial Meeting on Discourse and Dialogue, Seoul, South Korea.

Byron, D., A. Koller, K. Striegnitz, J. Cassell, R. Dale, J. Moore, and J. Oberlander. 2009. Report on the first NLG challenge on Generating Instructions in Virtual Environments (GIVE). Proceedings of the 12th European Workshop on Natural Language Generation (ENLG), Athens, Greece.

Calder, J., M. Reape, and H. Zeevat. 1989. An algorithm for generation in unification categorial grammar. Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Manchester, UK.

Chen, D. L., J. Kim, and R. J. Mooney. 2010. Training a multilingual sportscaster: using perceptual context to learn language. Journal of Artificial Intelligence Research 37. 397–435.

Cohen, P., and C. R. Perrault. 1986. Elements of a plan-based theory of speech acts, 423–440. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Cuayáhuitl, H. 2009. Hierarchical reinforcement learning for spoken dialogue systems. PhD Thesis, University of Edinburgh, Scotland, School of Informatics.

Dale, R. 1989. Cooking up referring expressions. Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL), Vancouver, British Columbia, Canada.

Demberg, V., and J. Moore. 2006. Information presentation in spoken dialogue systems. Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Trento, Italy, 353–360.

Denis, A. 2010. Generating referring expressions with reference domain theory. Proceedings of the 6th International Natural Language Generation Conference (INLG), Dublin, Ireland.

Dethlefs, N., and H. Cuayáhuitl. 2010. Hierarchical reinforcement learning for adaptive text generation. Proceedings of the 6th International Natural Language Generation Conference (INLG), Dublin, Ireland.

——. 2011. Hierarchical reinforcement learning and hidden Markov models for task-oriented natural language generation. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT), Portland, Oregon, USA.

——, Y. Wu, A. Kazerani, and S. Winter. 2011. Generation of adaptive route descriptions in urban environments. Spatial Cognition and Computation 11(2) 153–177.

——, H. Hastie, V. Rieser, and O. Lemon. (2012). Optimising incremental dialogue decisions using information density for interactive systems. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Jeju, South Korea.

DeVault, D., D. Traum, and R. Artstein. 2008. Practical grammar-based NLG from examples. Proceedings of the 5th International Conference on Natural Language Generation (INLG), Salt Fork, Ohio, USA.

Domingos, P. 2012. A few useful things to know about machine learning. Communications of the ACM 55(10). 78–87.

Elhadad, M. 1993. Using a regimentation to control lexical choice: a functional unification-based approach. PhD Thesis, Department of Computer Science, Columbia University, New York.

Foster, M. E., and J. Oberlander. 2006. Data-driven generation of emphatic facial displays. Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Trento, Italy, 353–360.

Gargett, A., K. Garoufi, A. Koller, and K. Striegnitz. 2010. The GIVE-2 corpus of generating instructions in virtual environments. Proceedings of the 7th International Conference on Language Resources and Evaluation, Malta.

Garoufi, K., and A. Koller. 2010. Automated planning for situated natural language generation. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden.

Golland, D., P. Liang, and D. Klein. 2010. A game-theoretic approach to generating spatial descriptions. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), MIT, Massachusetts, USA.

Heeman, P., and G. Hirst. 1995. Collaborating on referring expressions. Computational Linguistics 21. 351–382.

Hovy, E. 1991. Approaches to the planning of coherent text. Natural language generation in artificial intelligence and computational linguistics, ed. by C. Paris, W. Swartout and W. Mann, 83–102. Boston, Massachusetts: Kluwer Academic Publishers.

Janarthanam, S., and O. Lemon. 2010. Learning to adapt to unknown users: referring expression generation in spoken dialogue systems. Proceedings of the 48th Annual Meeting of the Association of Computational Linguistics (ACL), Uppsala, Sweden.

Jordan, P., S. Katz, P. Albacete, M. Ford, and C. Wilson. 2012. Reformulating student contributions in tutorial dialogue. Proceedings of the 7th International Conference on Natural Language Generation (INLG), Chicago, IL, USA.

Joshi, A. 1987. the relevance of tree adjoining grammar to natural language generation. Natural language generation, ed. by G. Kempen. Dordrecht, The Netherlands: Martinus Nijhoff Press.

Kay, M. 1985. Unification in grammar. Natural language understanding and logic programming, ed. by V. Dahl and P. Saint-Dizier. Amsterdam: North-Holland.

Koller, A., and R. Petrick. 2011. Experiences with planning for natural language generation. Computational Intelligence 27(1) 23–40.

——, K. Striegnitz, D. Byron, J. Cassell, R. Dale, and J. Moore. 2010. The first challenge on generating instructions in virtual environments. Empirical methods in natural language generation, ed. by M. Theune and E. Krahmer, 337–361. Berlin/Heidelberg: Springer Verlag.

Langkilde, I., and K. Knight. 1998. Generation that exploits corpus-based statistical knowledge. Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics.

Lavoie, B., and O. Rambow. 1997. A fast and portable realizer for text generation systems. Proceedings of the Fifth Conference on Applied Natural Language Processing, Washington, DC.

Lemon, O., P. Parikh, and S. Peters. 2002. Probabilistic dialogue modelling. Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue, Philadelphia, USA.

Levin, E., R. Pieraccini, and W. Eckert. 2000. A stochastic model of computer-human interaction for learning dialogue strategies. IEEE Transactions on Speech and Audio Processing 8. 11–23.

Mahamood, S., and E. Reiter. 2012. Working with clinicians to improve a patient-information NLG system. Proceedings of the 7th International Conference on Natural Language Generation (INLG), Chicago, IL, USA.

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

McDonald, D. D. 1976. Natural language production as a process of decision-making under constraints. PhD thesis, Massachusetts Institute of Technology (MIT).

McKeown, K. R. 1985. Text generation: using discourse strategies and focus constraints to generate natural language text. Cambridge: Cambridge University Press.

Mel'cuk, I. 1988. Dependency syntax: theory and practice. New York: State University of New York Press.

Mitchell, T. 1997. Machine learning. New York: McGraw-Hill.

Moore, J. D., and C. L. Paris. 1994. Planning text for advisory dialogues: capturing intentional and rhetorical information. Computational Linguistics 19. 651–694.

Neches, R., W. Swartout, and J. Moore. 1985. Enhanced maintenance and explanation of expert systems through explicit models of their development. IEEE Transactions on Software Engineering 11(11). 1337–1351.

Oh, A. and A. Rudnicky. 2000. Stochastic language generation for spoken dialogue systems. Proceedings of the ANLP/NAACL workshop on conversational systems, Seattle, Washington, USA, 27–32.
Paris, C. L. 1993. User modelling in text generation. London: Pinter Publishers.
Patten, T. 1988. Systemic Text generation as problem solving. Cambridge, England: Cambridge University Press.
Pietquin, O., M. Geist, S. Chandramohan, and F.-B. Herve. 2011. Sample-Efficient Batch Reinforcement Learning for Dialogue Management Optimisation. ACM Transactions on Speech and Language Processing (Special Issue on Machine Learning for Robust and Adaptive Spoken Dialogue Systems) 7(3) 1–21.
Piwek, P., and K. van Deemter. 2007. Generating under global constraints: the case of scripted dialogue. Research on Language and Computation 5(2) 237–263.
Ratnaparkhi, A. 2000. Trainable methods for surface natural language generation. *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, 194–201.
Reiter, E. 1994. Has a consensus NL generation architecture appeared and is it psycholinguistically plausible? *Proceedings of the 7th International Workshop on Natural Language Generation*.
——, and R. Dale. 2000. Building natural language generation systems. Cambridge: Cambridge University Press.
——, and C. Mellish. 1992. Using classification to generate text. Proceedings of the 9th International Conference on Computational Linguistics (COLING), Nantes, France.
Rieser, V., O. Lemon, and X. Liu. 2010. Optimising information presentation for spoken dialogue systems. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL), Uppsala, Sweden.
Roth, M., and A. Frank. 2010. Computing EM-based alignments of routes and route directions as a basis for natural language generation. Proceedings of the 23rd International Conference on Computational Linguistics (Coling), Beijing, China.
Skantze, G., and A. Hjalmarsson. 2010. Towards incremental speech generation in dialogue systems. Proceedings of the 11th Annual Meeting on Discourse and Dialogue (SIGdial), Tokyo, Japan.
van der Sluis, I., and E. Krahmer. 2007. Generating multimodal referring expressions. Discourse processes. Special Issue on “Dialogue Modelling: Computational and Empirical Approaches” 44(3) 145–174.
Stalnaker, R. 1998. On the representation of context. Journal Logic, Language and Information 7(1) 3–19.
Stent, A., S. Bangalore, and G. D. Fabbrizio. 2008. Where do the words come from? Learning models for word choice and ordering from spoken dialog corpora. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).
Stoia, L., D. Shockley, D. Byron, and E. Fosler-Lussier. 2006. Noun phrase generation for situated dialogs. Proceedings of the 4th International Conference on Natural Language Generation (INLG), Sydney, Australia.
Stone, M., and C. Doran. 1996. Paying heed to collocations. Proceedings of the 8th International Workshop on Natural Language Generation, Herstmonceux Castle UK.
Sutton, R. S., and A. G. Barto. 1998. Reinforcement learning: an introduction. Cambridge, Massachusetts: MIT Press.
Viethen, H. A. E. 2011. The generation of natural descriptions: corpus-based investigations of referring expressions in visual domains. PhD Thesis, Macquarie University, Sydney, Australia.
Walker, M., D. Litman, C. Kamm, and A. Abella. 1997. PARADISE: a framework for evaluating spoken dialogue agents. Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL), Madrid, Spain.
——, A. Stent, F. Mairesse, and R. Prasad. 2007. Individual and domain adaptation in sentence planning for dialogue. Journal of Artificial Intelligence Research (JAIR) 30. 413–456.
Ward, N. 1994. A connectionist language generator. Norwood, New Jersey: Ablex.
White, M., R. Clark, and J. Moore. 2010. Generating tailored, comparative descriptions with contextually appropriate intonation. Computational Linguistics 36(2) 159–201.
Witten, I. H., and E. Frank. 2005. Data mining: practical machine learning tools and techniques. San Francisco: Morgan Kaufman.