A dialogue-based knowledge authoring system for text generation

Alistair Knott and Nick Wright
Dept of Computer Science, University of Otago*

January 22, 2003

Abstract

One of the main obstacles to the development of any text generation system is the difficulty of creating the knowledge base from which texts are generated. In particular, a system which performs sentence generation 'from scratch' requires a great deal of information about how concepts in its knowledge base are expressed syntactically. Unsurprisingly, it has proven very difficult to build tools which acquire such information from people with no specialist expertise in linguistics and semantics. However, there is another method for acquiring such information which has been largely overlooked: if we want to build a knowledge base that allows full sentence generation, all we need is a *bidirectional* sentence processing system in which a single declarative grammar supports both sentence generation and sentence interpretation. With such a system, authoring new facts in the knowlege base can be achieved very naturally in the context of a natural language dialogue.

1 Introduction

Natural language generation (NLG) systems are approaching a point where the quality of their *output* is sufficient for commercial applications. For instance, systems for the generation of personalised multimedia/web pages (Bateman *et al.*, 2001; O'Donnell *et al.*, 2001), of technical documentation (Reiter *et al.*, 1995) and of explanation of complex concepts (Lester and Porter, 1997) can produce reasonably high-quality texts, which quite reliably fulfil their intended communicative functions. What stands in the way of the commercial development of such systems is not primarily the quality of their output, but the difficulty of producing the databases which serve as their *input*. As noted by Kittredge *et al.* (1991), this database must contain not only 'domain knowledge' about the application domain itself, but also 'domain communication knowledge' about how this information is communicated linguistically—in particular, how concepts in the domain are expressed syntactically. Both kinds of information are very difficult to elicit from domain experts without a technical background in knowledge representation (KR) and linguistics. In fact, the problem of authoring the database of an NLG system is widely regarded as the main obstacle to the development of significant commercial NLG applications—see for instance Reiter and Dale (1997), Power and Scott (1998), Reiter (2000).

In this paper, we begin in Section 2 by surveying some current proposals for addressing or minimising the knowledge acquisition problem in NLG. In Section 3, we present and re-evaluate a very simple alternative proposal—namely that a NLG system's knowledge base can be authored using natural language sentences. This proposal is typically disregarded as a method of authoring knowledge bases, because it makes unrealistic demands on the sentence interpretation module. While this seems true for knowledge base authoring in general, we argue that the knowledge base of a NLG system is a special case for which text-based knowledge acquisition is a more realistic prospect. In Section 4, we describe a prototype knowledge authoring system built to test this suggestion. We conclude in Section 5 with a summary.

2 Approaches to knowledge acquisition in NLG

There are several possible strategies for addressing problems with knowledge acquisition in NLG. Four

*This work was supported by University of Otago Research Grant MFHB10, and by the NZ Foundation for Research in Science & Technology grant UOOX02.

recently-pursued strategies will be considered in this section.

2.1 Template-based systems

There has been a strong trend recently towards developing NLG systems that operate using **templates** or 'canned text' rather than by generating sentences from scratch. In such systems, information is not only expressed linguistically, it is also *stored* linguistically. A template is basically a natural language sentence, with some 'slots' in it which can be filled differently in different contexts. Typically there are slots for some or all of the referring expressions in the sentence, which are filled by a specialised referring expression generation module. There are typically also slots for agreement morphology elsewhere in the sentence, so that this is kept consistent with the referring expressions selected.

A central benefit of templates over sentence generation systems is that they are easier to create and use than fully-fledged generation grammars. Reiter (1999) notes, for instance, that the documentation for many existing sentence generation systems is quite sparse, and therefore that any problems with syntactic coverage are difficult to remedy. On the other hand, Hirst *et al.* (1997) describe a system for authoring template-like structures which has been used with some success by domain experts with no technical knowledge about NLG.

The main disadvantage of template methods is that a great deal of flexibility and generality is lost. For instance, in different contexts we might want to vary the tense, aspect, mood, and information structure of a sentence being generated; if we are using templates, each sentence will probably have to be represented using several different templates in order to achieve this. (A good discussion of the benefits and disadvantages of templates is given in Reiter, 1999).

2.2 Knowledge base transfer

An alternative approach to knowledge acquisition is to adapt an existing knowledge base in some given domain to serve as input to a NLG system. Typically the knowledge base will not contain any domain communication information, so the main work is in specifying suitable syntactic structures (and possibly discourse structures) for conveying different concepts in the knowledge base.

Many useful sources of knowledge can be tapped in this way. A good example is the Lester and Porter's (1997) KNIGHT system, which produces explanations of concepts in biology using a very large and rich knowledge base not originally designed with NLG in mind. At the other end of the spectrum, O'Donnell *et al.* (2001) describe a system which generates descriptions of objects in a musem, whose database is (in part) parsed directly from the museum's own electronic catalogue. Records in this catalogue contain some numerical and some textual fields, which are translated into semantic and template-based information respectively.

Database conversion efforts such as these certainly constitute one useful method for constructing NLG knowledge bases. But it is likely that at least some portion of an NLG system's knowledge base still needs to be authored 'by hand', fact-by-fact, by a domain expert. So something else is also needed.

2.3 Knowledge acquisition protocols

Another proposed method for building an NLG knowledge base is to employ tried-and-tested knowledge-elicitation techniques from the field of expert systems. There are many similarities between the knowledge base used by an expert system for reasoning in a particular domain and the knowledge base used by an NLG system—in particular, both types of knowledge base make use of complex knowledge representation formalisms, and both serve to produce system output which is designed to be intelligible by a human without an understanding of these formalisms.

Reiter *et al.* (2000) describe a project to use structured knowledge-elicitation techniques in the development of an NLG system. Techniques tried included sorting, think-aloud protocols, and iterative revisions of generated texts. The verdict of this study was that while some of these techniques appeared useful in particular (rather domain-specific) respects, they were certainly 'not a panacea', and 'need to be used with some caution'.

2.4 Knowledge authoring systems

A final method for creating NLG knowledge bases is to build a special-purpose authoring tool. Ideally this tool should be easy to use by a domain expert, and to allow sophisticated KR structures to be created without demanding specialised knowledge. In other words, it should basically hide the full complexity of the underlying knowledge representation from the domain expert.

Several authoring aids have been developed for NLG systems. A common approach is to provide a

graphical interface to the system's KR structures—for instance, there are several sentence generators that provide a diagram-based editor for inspecting and/or building the typed feature hierarchies which they employ (see e.g. Bateman, 1997; Copestake, 2000). While these systems certainly make knowledge authoring easier for a person with the appropriate background in knowledge representation and linguistics, our own experience is that they are far from being useable by non-technical domain experts.

A more promising approach is taken by Power *et al.* 1998 (see also Power and Scott, 1998). The key intuition here is that the authoring tool for a NLG system should make use of natural language as its interface. But Power *et al.* foresee problems if domain authors are permitted to enter unrestricted free text. Instead, they propose a paradigm in which a general-purpose sentence is produced by the system, and progressively refined by the author changing words and phrases. Each refinement results in changes being made to the knowledge base from which the sentence is produced, after which a sentence reflecting the changed knowledge base is generated to be checked by the author. The resulting paradigm is rather neatly termed WYSIWYM ('What you see is what you mean') editing. This paradigm is certainly an attractive one, and takes a sensibly conservative line on the limits of current sentence interpretation sytems. However, in the remainder of this paper, we would like to question this conservatism.

3 NL dialogue for knowledge base authoring

Our central proposal is that the knowledge needed to build an NLG system can be acquired from the domain expert primarily by the expert entering sentences in free text, in the context of a man-machine dialogue. We believe that a dialogue system is in fact the most natural medium by which an expert with no technical knowledge of KR and linguistics can populate an NLG system's knowledge base. And we believe that the current generation of dialogue systems provide some measure of proof-of-concept for this approach.

Before considering the practicality of this idea, we will begin by considering its desirability. A natural-language-based authoring tool certainly seems an attractive proposition. Consider the case of a domain expert who wants to add a new fact to a system that generates descriptions of objects in a museum—perhaps the fact that a certain artefact (say the necklace in case 8) was designed by a certain designer (say Jessie King).[1] The expert would simply have to type in a sentence like the following:

(1) The necklace in case 8 was designed by Jessie King.

Assume now that this sentence is processed by a sentence interpretation system, which derives a semantic representation for it in a suitable logical formalism. Finally, assume that the sentence processing system is bidirectional; i.e. that the same grammar which was used by the interpretation system can also be used to transform logical representations back into sentences. The system will then have the ability to regenerate the original input sentence. What is more, small changes to the input knowledge representation reflecting different contexts will be able to generate many different variants on the input sentence, such as

(2) This necklace was designed by Jessie King.

(3) Jessie King designed it.

(4) The designer who designed the necklace in Case 8 [has some other property].

and so on. We suggest that *if* a sentence interpretation system is sufficiently robust in a given domain to determine suitable semantics for sentences entered by the domain expert, then this method of authoring will certainly be more practical than any of the methods considered in Section 2.

The possibility of an authoring system based on unrestricted natural language input has not been considered very seriously in the NLG literature. Those who do consider it (e.g. Power *et al.*, 1998) typically only do so to discard it in favour of a more constrained form of natural-language input. The idea of using natural language to author the knowlege base of an expert system was considered quite seriously by Pulman (1996), but again it was dismissed in favour of the use of a controlled language. In the remainder of this section, we will consider some of the obvious problems inherent in using unrestricted text input for knowledge acquisition. In each case, we will claim that these difficulties are surmountable for the particular case of NLG authoring.

[1] By considering this particular case, we continue a venerable tradition in NLG examples.

3.1 Lack of syntactic coverage

Current symbolic grammars for natural language all have limited syntactic coverage. This is particularly true for grammars which derive a detailed semantic representation of parsed sentences. There are a number of 'wide-coverage' symbolic grammars which perform semantic interpretation, but most of these grammars still fail to parse a large proportion of sentences in a naturally-occurring corpus.

However, it must be borne in mind that NLG systems tend to have restricted syntactic coverage anyway. Restricted coverage is far less of a problem for NLG systems than for interpretation systems. A generation system only needs to talk about a restricted range of things—namely the things which are in its domain. Maybe an existing wide-coverage grammar, with extensions to cover syntactic constructions specific to the domain of the generation sytem, would have wide enough coverage to allow natural language input in the domain in question. This at least seems like a possibility worth investigating.

3.2 Semantic accuracy

Another potential problem for a text-based authoring system concerns the semantic representations which are derived for input sentences. What means does the author have of checking that these representations really are the ones (s)he had in mind? The problem is certainly a serious one for an application like an expert system, where the representation derived will be used in a complex theorem-proving algorithm, in which the effects of a representation are hard to test systematically. However, in a NLG system we can be more concrete about what counts as an adequate representation: it simply has to be one from which an appropriate range of paraphrase sentences can be generated. If we imagine a context where any sentence entered by the domain expert can be immediately paraphrased by the system, the author has an instant check on whether or not the representation it computed was suitable.

3.3 Ambiguity

A final serious problem with wide-coverage sentence interpretation systems is the number of alternative interpretations which are computed for the average sentence. As is well known, there are several sources of ambiguity: some of the more serious ones include syntactic attachment ambiguities, lexical ambiguities, referential ambiguities and quantifier scope ambiguities. All these ambiguities multiply as the coverage of a grammar increases. How can these be dealt with?

To begin with, some of these ambiguities are not particularly significant for a NLG system. For instance, quantifier scope ambiguities are frequently not reflected at all in regenerated texts. Furthermore, many types of syntactic ambiguity are likely to be resolvable by consulting a statistical parser running in parallel with the sentence interpretation system. Statistical parsers (e.g. Collins, 1996) are quite good at eliminating very low-probability parses of sentences.

For the remaining ambiguities, we suggest that a system of followup or clarification questions may be sufficient to elicit from the author which is the intended reading. In the case of referential ambiguities, for instance, the system could simply ask the author to further clarify which object is being referred to. In the case of syntactic or semantic ambiguities, the system could simply generate a sentence (or a set of sentences) to realise each of the alternative interpretations, and ask the author to select which is the correct one. (If, as sometimes happens, some of the alternative interpretations result in the same set of sentences being generated, then it doesn't really matter from our point of view which interpretation is selected.)

In summary, we believe that it is possible to plan a fairly general set of followup questions which stand a good chance of nailing many of the attendant ambiguities of unrestricted natural language input. The upshot of this is firstly that a text-based authoring system does seem at least a possibility, and secondly that any such system should take the form of a *dialogue* between the author and the system, in which clarification questions from the system play an important role.

4 A dialogue based knowledge authoring tool

We have built a prototype dialogue system to investigate the feasibility of natural-language-based knowledge base acquisition. The prototype is an extension of a system called Te Kaitito, a collection of bilingual NLP modules for sentence processing, sentence translation and dialogue applications in English and M¯aori (see e.g. Knott *et al.*, 2002; Bayard *et al.*, 2002).

4.1 Overview of Te Kaitito's dialogue system

In one mode, Te Kaitito functions as a dialogue-based natural language interface to a database of facts. Facts can be added to the database using declarative sentences in either English or M¯aori, and queries of the database can be made using questions in either language. The system responds with acknowledgements and/or answers when appropriate; it is also able to ask a range of follow-up questions in cases where there are problems interpreting an incoming sentence or question.

Te Kaitito uses the LKB system (Copestake, 2000) as its grammar development environment. LKB is a bidirectional sentence processing system: the same declarative grammar is used by the sentence interpreter and the sentence generator, so any sentence which the system can interpret, it is also able to generate. The semantic representation into which sentences are interpreted uses a formalism called Minimal Recursion Semantics (MRS—see Copestake *et al.*, 1999). MRS is a rich semantic language with support for useful features like underspecification and scope ambiguity, and is well suited to both sentence interpretation and generation.

For the discourse and dialogue components of Te Kaitito, MRS representations are converted to Discourse Representation Structures (DRSs). The current discourse context is stored as a **context DRS**, comprising a set of **DRS referents** (the objects which have been mentioned in the discourse so far) and a set of **conditions** (the properties which have been asserted about these referents). When a sentence is processed by Te Kaitito, the system first computes its MRS representation, and then converts this into a DRS, comprising an **assertion** and a set of **presuppositions**, roughly along the lines of Van der Sandt (1992). The system then attempts to resolve these presuppositions using the context DRS. If it fails, it responds to the user with a clarification question. If it succeeds, then if the sentence is declarative, it adds the appropriate new material to the context DRS, and if it is a question, it searches the context DRS to generate an answer. More details of Te Kaitito's dialogue manager can be found in de Jager *et al.* (2002).

When Te Kaitito generates a sentence in response to the user (whether it is a question or an answer), an initial process of **sentence planning** produces a complete MRS, and this MRS is then sent to LKB's generator, to produce a sentence in an appropriate language. Currently, the main task in sentence planning is deciding on appropriate **referring expressions**. For instance, when the system provides the answer to a *wh*-question, the referring expression it uses must distinguish the entity referred to from all other entities in the context. We currently use an algorithm based on that given in Dale and Reiter (1992) to plan referring expressions.

4.2 An architecture for dialogue-based knowledge authoring

The dialogue system as it stands clearly allows a human user to create a knowledge base of facts through a natural language dialogue: the 'knowledge base' is the context DRS, which is updated as the dialogue proceeds. However, several modifications need to be made to the system as outlined above to create the kind of knowledge base which can serve as the input to a NLG system. There are four problems to address.

Problem 1 The context DRS does not contain any information about which sets of DRS conditions are realisable as single sentences. It is basically a flat list of DRS conditions without any structure. One of the tasks in a generation system is to choose sets of facts from which individual sentences are to be built; the current context DRS does not contain any information to guide this decision.

Problem 2 The context DRS does not contain information in the format needed by the sentence generator. The generator uses MRS representations, not DRS conditions.

Problem 3 The sentences which are entered by the user will almost certainly reflect the particular dialogue *context* in which they were entered. For instance, their referring expressions are likely to be tailored to the context; their tense and aspect might likewise be contextually determined; or they might contain sentence/clause connectives linking back to propositions in the context. The knowledge base for a NLG system needs to allow a fact to appear in a range of different discourse contexts. It really needs to be able to store *facts* rather than specific sentences.

Problem 4 The context DRS does not contain any information about how multisentence texts are to be planned. An authoring tool for an NLG system needs to provide a way of entering this information as well as information about individual sentences.

4.2.1 Fact nodes

To address Problem 1 above, we use an architecture in which the declarative sentences entered by the user result in extra structures being built specifically to support sentence generation. When an declarative sentence is entered, some new information is added to the context DRS, but in addition, a new element called a **fact node** is created and stored in a structure called a **knowledge graph**. Each fact node represents a 'clause-sized' piece of semantic content. The assumption is that the grouping of semantic material into sentences which is employed by the domain expert in his/her chosen sentences will be a good one for the NLG application to maintain. (Other features of the knowledge graph are described in Section 4.2.2.)

To address Problem 2, we use MRS structures to represent the content of fact nodes in the knowledge graph. Each time a declarative sentence is added to the context DRS, a new fact node is created in the knowlege graph, and the sentence's MRS structure serves as the basis for the creation of an appropriate MRS representation for this fact node.

To address Problem 3, we do not simply *duplicate* the MRS of the incoming sentence in a newly-created fact node. We remove any portions of the MRS structure which relate to context-specific material in the original sentence, and replace them with context-independent representations. In particular, we remove any portions of the incoming MRS which derive from context-sensitive referring expressions, and replace them with *pointers* to appropriate entities in the context DRS. When a sentence is to be generated from the fact node, these pointers are used by the referring expression planner to build new portions of MRS which are appropriate to the context in which generation is occurring. (MRS is a convenient formalism in this regard, because sentences are represented as relatively 'flat' structures, from which the relevant pieces can be easily unplugged.)

4.2.2 The knowledge graph

To address Problem 4 above, we need to ensure that the structures created to support generation permit not only the generation of single sentences, but also the planning of larger multisentence texts. To achieve this, we need to enrich the knowledge graph structure already described. The knowledge graph is actually a network with three kinds of node. Fact nodes hold sentence or clause-sized compilations of semantic information, as already mentioned. **Relation nodes** represent coherence relations between pairs of fact nodes, which can be used to plan the structure of multisentence texts. Finally **entity nodes** represent the entities in the knowledge base: each entity node is linked to all the fact nodes which contain a reference to this entity. The knowledge graph is essentially the same structure as the 'content potential' of O'Donnell *et al.*, 2001; see this paper for more details.

We would like to be able to author relation nodes using natural language input as well as fact nodes. We have not done much work in this area as yet. Currently we make some very simple assumptions; for instance, if the user enters two consecutive sentences which are aspectually events, then a SEQUENCE relation holds between them, while if a state is followed by an event, a BACKGROUND relation holds between them. These strategies are basically just placeholders for more serious work in rhetorical parsing.

4.2.3 User mode and authoring mode

A final extension which needs to be made is to distinguish between two modes in which a dialogue can occur. In **authoring mode**, the human interlocutor is the domain expert, setting up the knowledge base for an NLG system, and the system is facilitating this process. In **user mode**, the human interlocutor is the intended user of the generation system; i.e. the system is generating texts for the benefit of this user. To support this distinction, we need to create two DRS-based repositories of information; one to hold the system's private knowledge (termed the **global DRS**, and one to hold any common knowledge shared by the system and the interlocutor as a result of their current discourse (which we will continue to call the **context DRS**). During authoring mode, the two DRSs are basically identical, because everything the system knows it has been told during the current dialogue. If the system is put into user mode after an authoring dialogue, the global DRS is retained, and the context DRS is reset to empty. Consequently, note that the pointers to DRS entities in fact nodes which were mentioned in Section 4.2.1 must be to entities in the *global* DRS. However, the algorithm for creating referring expressions has to make reference to both the global DRS and the context DRS. (For instance, if the entity in question does not yet appear in the context DRS, an indefinite NP must be generated based on properties found in the global DRS.)

4.2.4 An example

Figure 1 gives a sample dialogue with the prototype system, illustrating a number of the features described above. The system starts off in authoring mode. In lines 1–10, the user (i.e. the 'domain expert') enters a numer of declarative sentences and the system acknowledges them. Note that one of these sentences (line 5) results in a clarification subdialogue (lines 6–8). The system is then switched to user mode and is asked by the user to generate a story about what it knows (lines 11–12). The system responds with a very simple multisentence text (line 13). This text is built from our (equally simple) knowledge graph, using the text planning algorithm described in O'Donnell *et al.* (2001). In this story, notice that sentences appear in a different order than they were entered by the domain author. Note also that as a result of this, the referring expressions in the sentences are different: some indefinite NPs have become definite, and vice versa. There are other differences in the referring expressions too: in particular, the adjective *green* used by the author to refer to the monkey is dropped in the generated text, because it is not needed to identify the monkey uniquely. Note finally that some sentences are generated with different syntactic structures than their originals; for instance, the author's sentence 9 is generated in the passive. (In fact, the complete story can also be rendered in M¯aori.)

1	User:	A big dog was in a garden.
2	Te K:	*Ok.*
3	User:	A tree was in the garden.
4	Te K:	*Ok.*
5	User:	The dog chased the monkey.
6	Te K:	*Which monkey did the dog chase?*
7	User:	A green monkey.
8	Te K:	*Ok. Ok.*
9	User:	The monkey chased a cat.
10	Te K:	*Ok.*
11	User:	[User mode.]
12	User:	[Generate story.]
13	Te K:	*Once upon a time, a tree was in a garden. A dog chased a monkey; then a cat was chased by the monkey. The dog was also in the garden. They all lived happily ever after.*
14	User:	What did the monkey chase?
15	Te K:	*It was the cat.*

Figure 1: A sample authoring dialogue

As a final observation, note that in lines 14 and 15, the user asks the system a question and receives an answer. An additional benefit of a bidirectional sentence-processing system in an NLG application is that *queries* can be made by the user to follow up on the texts (s)he is presented. This provides huge value-added for a generation system, putting its knowledge base to much more use, with practically no additional infrastructure. In fact, in our application, knowledge authoring, text generation and information-seeking dialogue are integrated quite seamlessly.

5 Summary and extensions

In this paper we have suggested that the recalcitrant problem of knowledge authoring in NLG can be addressed by developing bidirectional systems, which can interpret all the sentences they can generate. The knowledge base of a generation system can then be populated simply by entering sentences in natural language—the kind of sentences (but naturally not *exactly* the same sentences) as the ones the author wants the system to generate. This means of knowledge authoring is undoubtedly a tall order for current sentence interpretation systems. However, we argue that the interpretation task for NLG authoring is a special case, (a) because NLG systems require limited coverage, and (b) because methods exist for the author to check the representations derived by the sentence interpreter using clarification questions in a dialogue context. We predict that knowledge-authoring systems built using this kind of natural language input will be much more readily usable by non-technical domain authors than any of the existing range of authoring systems.

References

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

Bateman, J., Kamps, T., Kleinz, J., and Reichenberger, K. (2001). Constructive text, diagram and layout generation for information presentation: the DArt$_{bio}$ system. *Computational Linguistics*, **27**(3), 409–449.

Bayard, I., Knott, A., and de Jager, S. (2002). A unification-based grammar for a fragment of M¯aori and English. In *Proceedings of the 2nd*

Australasian Natural Language Processing Workshop (ANLP 2002).

Collins, M. (1996). A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Meeting of the ACL*, Santa Cruz.

Copestake, A. (2000). The (new) LKB system. CSLI, Stanford University.

Copestake, A., Flickinger, D., Sag, I., and Pollard, C. (1999). Minimal Recursion Semantics: An introduction. Manuscript, CSLI, Stanford University.

Dale, R. and Reiter, E. (1992). A fast algorithm for the generation of referring expressions. *Proceedings of COLING*.

de Jager, S., Knott, A., and Bayard, I. (2002). A DRT-based framework for presuppositions in dialogue management. In *Proceedings of the 6th workshop on the semantics and pragmatics of dialogue (EDILOG 2002)*, Edinburgh.

Hirst, G., DiMarco, C., Hovy, E., and Parsons, K. (1997). Authoring and generating health-education documents that are tailored to the needs of the individual patient. In A. Jameson, C. Paris, and C. Tasso, editors, *User Modeling: Proceedings of the Sixth International Conference, UM97*, pages 107–118. Springer, Wien New York.

Kittredge, R., Korelsky, T., and Rambow, O. (1991). On the need for domain communication knowledge. *Computational Intelligence*, **7**, 305–314.

Knott, A., Bayard, I., de Jager, S., and Wright, N. (2002). An architecture for bilingual and bidirectional nlp. In *Proceedings of the 2nd Australasian Natural Language Processing Workshop (ANLP 2002)*.

Lester, J. and Porter, B. (1997). Developing and empirically evaluating robust explanation generators: The KNIGHT experiments. *Computational Linguistics*, **23**(1), 65–101.

O'Donnell, M., Mellish, C., Oberlander, J., and Knott, A. (2001). ILEX: an architecture for a dynamic hypertext generation system. *Natural Language Engineering*, **7**.

Power, R. and Scott, D. (1998). Multilingual authoring using feedback texts. In *Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 98)*, Montreal, Canada.

Power, R., Scott, D., and Evans, R. (1998). What you see is what you meant: direct knowledge editing with natural language feedback. In *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI)*, Brighton, UK.

Pulman, S. (1996). Controlled language for knowledge representation. In *Proc. of the First International Workshop on Controlled Language Applications*, pages 233–242, Katholieke Universiteit Leuven, Belgium.

Reiter, E. (1997). NLG vs. templates. In *In Proceedings of the Fifth European Workshop on Natural-Language Generation (ENLGW-1995)*, Leiden, The Netherlands.

Reiter, E. (1999). Shallow vs. deep techniques for handling linguistic constraints and optimisations. In *Proceedings of the KI-99 Workshop on May I Speak Freely: Between Templates and Free Choice in Natural Language Generation*.

Reiter, E., Mellish, C., and Levine, J. (1995). Automatic generation of technical documentation. *Applied Artificial Intelligence*, **9**.

Reiter, E., Robertson, R., and Osman, L. (2000). Knowledge acquisition for natural language generation. In *Proceedings of the First International Conference on Natural Language Generation*, pages 217–224.

Van der Sandt, R. (1992). Presupposition projection as anaphora resolution. *Journal of Semantics*, **9**, 333–377.