A question-answering system for English and Māori

Alistair Knott[1], Ian Bayard[1], Samson de Jager[1], Lindsay Smith[1],
John Moorfield[2], Richard O'Keefe[1]

[1]Dept of Computer Science, [2]School of Māori Studies
University of Otago

Abstract

This paper describes Te Kaitito: a system for authoring and querying database information in English and Māori. The system is designed to serve as a general platform for training and research in natural language processing; the topics we are currently focussing on include the use of a bidirectional grammar in natural language generation; the use of discourse representation theory (Kamp and Reyle, 1993) as the semantic foundation for a text generation system, and the the development of a syntactic and semantic model of the Māori language.

Keywords: natural language processing, Māori, Discourse Representation Theory, text generation

1. Introduction

This paper describes Te Kaitito: a system for authoring and querying database information in English and Māori. (*Te Kaitito* is Māori for 'the composer', or 'the improviser'.) The system is intended to fulfil several functions. Firstly, it is intended to serve as a platform for developing computational models of syntax, semantics and discourse, and particularly as a training ground for students working in these areas. Secondly, it is intended to be the foundation for useful natural-language-processing applications, such as a natural language front-end for a database, a simple sentence translator, and a language-teaching tool. Finally, it is intended to act as a 'shop window' for computational linguistics techniques in the New Zealand community. A focus on the Māori language is particularly important in this regard; New Zealand is a bilingual country, and it is important to develop natural language processing resources for the indigenous language as part of the effort to ensure its survival.

We will begin in Section 2 by outlining the theoretical criteria which we are attempting to satisfy in the design of the system. In Section 3 we describe the system's architecture. Section 4 gives an example of a dialogue with the system.

2. Motivations for the system

Our initial focus was on building a text generation system for use in a web-based application such as dynamic hypertext (see e.g. Dale *et al.*, 1998; O'Donnell *et al.*, 2001). When the system was being designed, we focussed on a number of criteria which we felt would be useful for such a system.

2.1. Bidirectionality

The text generation research community has not been overly concerned with the issue of bidirectionality or reversibility. Text generation systems where the primary research interest is in generation frequently only perform generation. Systems which perform both generation and interpretation (e.g. Shieber, 1988; Van Noord, 1993) are typically built for use in machine translation applications, where many components of the generation process are simply not required. The generation module of a machine translation system only needs to be concerned with the linguistic realisation of a sentence-sized semantic message: there is no need to consider how and why this message is itself generated, and how the message in question is integrated into a larger discourse being produced.

However, there are good reasons for wanting a generation system to support sentence interpretation too. Our reasons both stem from the complexity of the knowledge bases required to support a generation system.

Firstly, in any environment in which text is being automatically generated, it would be useful to include a facility for question-answering. A generation system requires as input a knowledge base of facts in some computer-tractable representation. Given this fact, the extra effort of building a system that takes a query for this representation and generates a response should be relatively small. Likewise, a generation system requires a grammar, and a compositional mapping between grammatical structures and the semantic structures of the database representation. The extra effort of ensuring that the grammar is bidirectional should again be relatively small. On the other hand, the benefits of having a generation system which responds to natural language queries seem quite considerable. A generation system has to be able to respond flexibly to user input, but without sentence interpretation this input is typically very constrained, amounting in many cases simply to a set of menu options. In such cases, the poverty of the interface is often a limiting factor in the performance of the generation system, especially one which is designed to handle a sophisticated semantic representation at input.

Secondly, the performance of a generation system is also currently limited by the quality of the methods available for authoring its knowledge base. It is well known that knowledge base authoring is a bottleneck for current generation systems (see e.g. Paris, 2001). If it were possible to author a knowledge base by entering natural language sentences, this would certainly be a useful facility. Clearly, a constrained interface of some kind would be necessary to overcome problems relating to the interpretation of free text; however, there has already been a certain amount of work in this area (e.g. Power *et al.*, 1998; Piwek *et al.*, 1999).

If a system is going to support both generation and interpretation, we suggest it makes sense to use a bidirectional grammar, rather than developing specialised and independent generation and interpretation modules. See Neumann (1994) for a good summary of the advantages of bidirectional systems.

2.2. Multilinguality

From the point of view of effort versus reward, it makes sense for generation systems to target applications in which documents are to be produced in several languages (see e.g. Dale and Reiter, 2000). Given this fact, it makes sense to include a multilingual capability from the outset when a system is being developed, to avoid building language-specific assumptions into its design. Moreover, to ensure a good degree of language independence, it makes sense to aim for coverage of a set of languages which are not closely related to one another. While these are certainly not the main reasons why we are focussing on English and Māori, it is useful from this perspective that they are entirely unrelated languages (see Knott *et al*, 2001 for some illustrations).

A second decision we took was to incorporate our coverage of English and Māori into a single declarative grammar. The reason for this is partly theoretical parsimony and partly practical efficiency. From a theoretical perspective, we are interested in capturing the linguistic constraints which the languages share. From a practical perspective, we want to make it easy to add new languages which are syntactically similar to Māori (in particular Tongan and Samoan), and therefore want to develop from the outset a framework in which the similarities between two languages can be made explicit.

2.3. Semantics

Work in the semantics of natural language is not made use of as much as it could be in natural language generation. The semantics of noun phrases is an interesting case in point. Generation theorists tend to think of noun phrases as referring to objects in the world. Semanticists prefer to think of nouns in set-theoretic terms, within a general framework in which sentences are analysed as tripartite quantification structures featuring a bound variable, a 'restrictor set' and a 'nuclear scope set' (Barwise and Cooper, 1981; Heim, 1982). The determiner of a subject noun phrase introduces the bound variable and an appropriate quantification operator, the subject N-bar contributes the restrictor set, and the VP contributes the nuclear scope set. This approach seems madly complex for simple sentences, but pays off in the analysis of quantified sentences. There is a great deal of work on the generation of noun phrases (see e.g. Dale, 1988; Horacek, 1996), but the models proposed are not typically expressed in this set-theoretic vocabulary; and it is probably no coincidence that (with the exceptions of Creaney, 1996 and Power, 1999) there has not been much work on the generation of quantified sentences.

There are other topics in semantics which have likewise received little attention from generation researchers—for instance, there are few generation systems that explore the topic of presuppositions in great detail. As a general research focus, we are interested in exploring whether there are any good reasons for using analyses taken from semantic theory within a text generation system.

3. Overview of Te Kaitito

Figure 1 gives an overview of the architecture of Te Kaitito. Rounded boxes denote processing modules; square boxes denote data; solid arcs denote movement of data during processing for a conversational turn, and dashed arcs denote update of data structures after a turn.

The system operates via a web interface. The user types in a sentence, which is parsed using an HPSG-style grammar of English and Māori. If no parse can be generated, an appropriate error message is returned directly to the user. If the parse succeeds, a set of **syntactic analyses** are generated, and passed to a semantic processor, which disambiguates between these alternatives, and then further disambiguates any remaining semantic ambiguities in the chosen sentence (relating mainly to quantifier scope and high-level concepts). The result of this process is a disambiguated semantic representation called a **scoped MRS structure**. (The notion of an MRS structure is explained in Section 3.2.) The scoped MRS may contain presuppositions, and so is passed to a presupposition resolution module. This module draws on a **discourse DRS** and a **salience list** (both defined in Section 3.3). Problems resolving presuppositions result in various kinds of follow-up question being formulated, with appropriate commands being passed to the sentence generation module. If presuppositions are adequately resolved, the resulting structure (termed the **sentence DRS**) is passed to the model checker. If the input sentence is a question, model checking involves computing an answer to the question; if it is a statement, then model checking involves checking the consistency of the information it contains with information already in

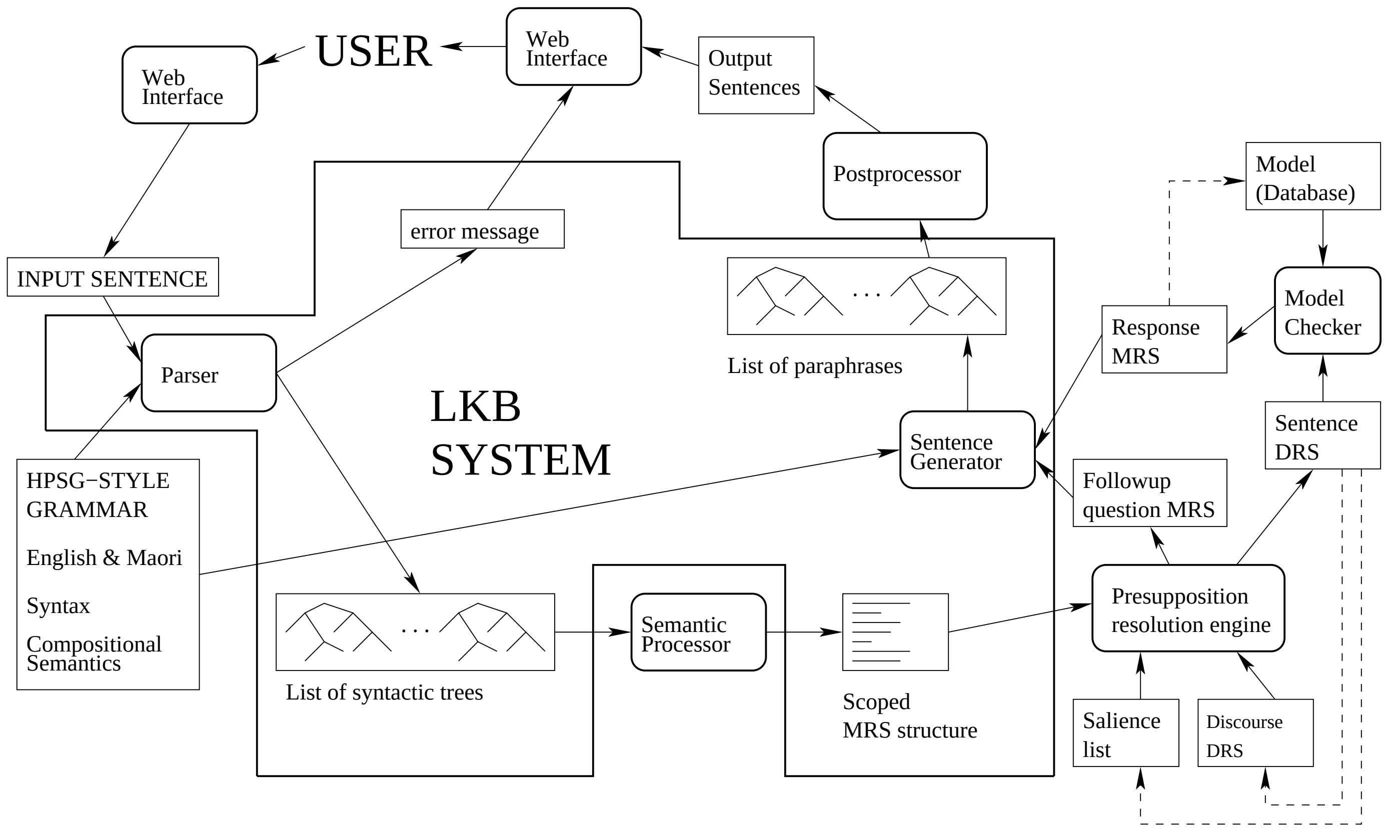

Figure 1: Architecture of Te Kaitito

the model. Questions are essentially a way of querying a database, and statements are a way of authoring a database. As well as generating a response, the system generates appropriate updates to some of its information structures: the discourse DRS, the salience list, and the model itself.

Figure 1 is in a few places guilty of wishful nomenclature. Some modules are little more than placeholders; in particular, the 'semantic processor' at present chooses at random beteen syntactic analyses, and at random between alternative quantifier scopings for the chosen analysis, and the responses generated from the presupposition and model checkers are currently just canned text rather than input structures for the sentence generator. Moreover, not all of the modules shown are currently integrated into our web-based system. At present, our web-based system doesn't consult a model, but simply paraphrases the user's initial sentence by interpreting it, deriving its semantic representation and then generating a set of sentences that realise this semantics in both English and Māori. With these caveats, we will describe the system as it is envisaged in the diagram, to emphasise how we intend to develop it as well as what we have currently done.

3.1. Syntax

For the syntactic construction we use the LKB (Linguistic Knowledge Building) system developed at Stanford and Cambridge (Copestake *et al*, 2000). This system is bidirectional, permitting sentence interpretation and sentence generation from the same declarative grammar. It also uses an interesting formalism for sentence semantics (see Section 3.2). The system itself is grammar independent but we have used an HPSG style grammar (Pollard and Sag, 1994). See Knott *et al* (2001) for details of our semantic treatment of English and Māori.

Our grammars for English and Māori are combined into a single grammar, with specific words and constructions from a given language being identified by values on a `language` feature. A number of general HPSG rules are language-independent (for instance, the Head Feature principle and the Head-Complement rule); the `language` feature is unspecified for such rules. We also leave the `language` feature unspecified for proper names. Agreement requirements on the `language` feature make it impossible to parse or generate sentences containing a mixture of words from different languages. One useful feature of a combined grammar of this sort is that our system can parse sentences in either language, without having first to identify which language they are in.

3.2. Semantics

The semantic formalism used by LKB is called **Minimal Recursion Semantics (MRS)** (Copestake *et al.,* 2001). The main characteristics of MRS are that it is designed to be compatible with feature-based grammar formalisms such as HPSG, that it provides a 'flat' (and therefore tractable) semantic representation for generation systems, and that it permits underspecified semantic representations. See Knott *et al* (2001) for more details about MRS representations and how we use them.

3.3. Presupposition module

The presupposition module has two tasks. Its first task is to turn the MRS representation for a sentence into a representation which makes its discourse characteristics more explicit, namely a discourse representation structure or **sentence DRS** (Kamp and Reyle, 1993) with presuppositions represented as a special kind of sub-DRS (van der Sandt, 1992). MRS representations are not exactly DRSs, because they do not associate sets of referents with the scopal elements introduced by quantifiers, and they do not represent DRT's notion of the accessibility relations between sub-DRSs in a complex DRS. Moreover, there is no explicit distinction between the presupposed elements of an MRS and the asserted elements. However, all of these elements can be retrieved from an MRS.

When a DRS-with-presuppositions has been created, its presuppositions (including ordinary anaphoric expressions) are resolved to create a **resolved DRS**, using a **discourse DRS** and a **saliency list** of referring expressions, both of which have been built up since the start of the discourse. (The saliency list currently contains a list of referents in order of decreasing recency, tagged with number and gender.) The presupposition module first finds all possible ways of resolving the DRS's presuppositions in the discourse DRS. If there are none, its output is used to generate an appropriate follow-up utterance. If there is more than one, the saliency list is consulted. If there is a sufficient difference between the saliency of the top candidate and the next candidate, the presupposition can be resolved automatically; othewise its output is used to generate a different type of follow-up utterance (along the lines of *Which X?*).

3.4. Model checker

When a sentence's presuppositions have all been resolved, it is converted to a **resolved DRS**, which is a DRS whose referents are either variables (for questions) or individuals in the model (in the latter case, either new or old). This structure is then compared to the **model**, which is the system's database. If the sentence was a question, the sentence DRS functions as a query to the model, and the response is either failure, or success, with a set of alternative sets of variable bindings. If the sentence was a statement, the model is updated with the new information. (We currently have no provision for checking the consistency of the newly-added information, but it is an extension we would like to make.)

3.5. Generation system

We use LKB's sentence generation system without modification. The input to the system is an MRS representation. The system operates using a form of lexically-driven chart generation, in which a chart is initialised with a set of words which realise individual EPs or groups of EPs, and all possible ways of creating sentences which use up all the EPs in the MRS (with appropriate bindings of their variables) are explored. The main inefficiency in this process is due to the presence of lexical items with 'null semantics' (for instance, the English copula in our semantic analysis) Any such words might be needed for syntactic reasons in the sentence to be created, but adding them all increases the complexity of the algorithm. The solution is a system of 'filter rules' that specify more tightly when each given item with null semantics will or might appear. See Carroll *et al.* (1999) for details of the generation algorithm.

With our current grammar, generation is very slow, ranging from around 5 seconds for simple sentences to over 30 seconds for complex ones involving several nested relative clauses. This is partly because the system looks for all the ways of realising a given MRS; allowing just a single solution would improve efficiency at least in some cases. Efficiency could also be improved by constraining the words which initialise the chart to only be of one language. Aside from these optimisations, improving the efficiency of the system is mainly a matter of improving our filter rules.

3.6. Web interface

The web interface is currently very simple. Te Kaitito is a Lisp process which runs continuously on the server machine; user sentences are picked up by a CGI script, which communicates with the Lisp system via a socket interface. The system processes each user sentence in turn and sends it back to be served back to the appropriate user. When a model, discourse DRS and salience list need to be stored for each user, this mechanism will have to be made more complicated; we envisage keeping the single-process model, and creating individual files for each user to store the relevant context in between turns.

4. A sample dialogue with Te Kaitito

In this section, we will give an example of a dialogue with Te Kaitito, in which the user creates a small knowledge base by typing in declarative sentences and then types in a question to query the knowledge base which was created. The example shows how the user's sentences are sensitive to the discourse context in which they appear. The dialogue is given in Figure 2. Note that

0	User	`[reset-model]`
1	User	`The greenstone fish-hook is in Case 8.`
	TK	`Okay.`
2	User	`A wooden fish-hook is in Case 6.`
	TK	`Okay.`
3	User	`The greenstone fish-hook has a leather strap.`
	TK	`Okay.`
		`...`
4	User	`Does the fish-hook which is in Case 8 have a strap?`
	TK	`Yes.`

Figure 2: A sample dialogue with Te Kaitito

the dialogue could have switched to Māori at any point, either when information was being added or queried.

After Step 0, the discourse DRS and the model are both initialised to empty. After Step 1, the sentence DRS and the resolved sentence DRS are given in Figures 3 (i) and (ii) respectively. DRS referents are given in lower case and model individuals are capitalised. (We omit tense and aspect information, for clarity.) In

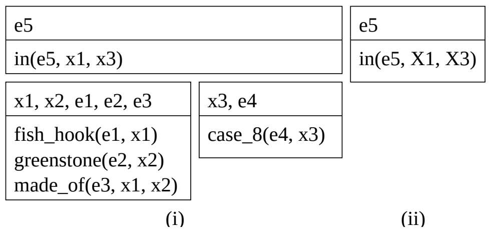

Figure 3: DRS and resolved DRS after Step 1

the sentence DRS, the upper box represents what the sentence asserts, and the lower boxes represent what the sentence presupposes. We use a separate presupposition box for each presuppositional expression in the sentence (except for expressions nested inside each other). This is to facilitate the generation of system responses in cases where a presupposition cannot be resolved for one reason or another. The presuppositions in this case cannot be resolved, because the model is initially empty. However, since the sentence is declarative, we are allowed to **accommodate** its presuppositions: the presupposed material is simply added to the model.[1] Using the updated model, we create the resolved DRS. This describes the material asserted by the sentence after its presuppositions have been resolved. The asserted

[1]If the sentence is a question, accommodation is not permitted.

material can involve new discourse referents (in our case just the event referent $e5$) and new predicates (in our case the predicate $in/3$). The resolved DRS is then **merged** with the model. Merging involves substituting new model individuals for any new discourse referents, adding these to the set of model individuals, and adding all the predicates in the resolved DRS to the model. The system responds with `Okay` when merging is complete.

Step 2 proceeds in a similar way; See Figure 4 (i) and (ii). This time the subject DP is not presupposed, so

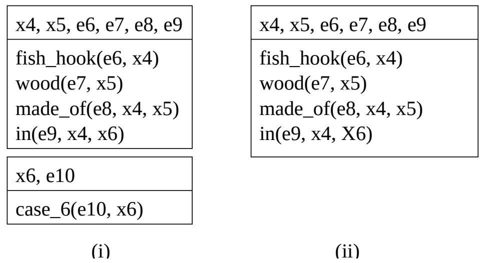

Figure 4: DRS and resolved DRS after Step 2

appears in the top box of the sentence DRS.

The results of Step 3 are shown in Figure 5 (i) and (ii). Note here that the presupposed material in the sentence

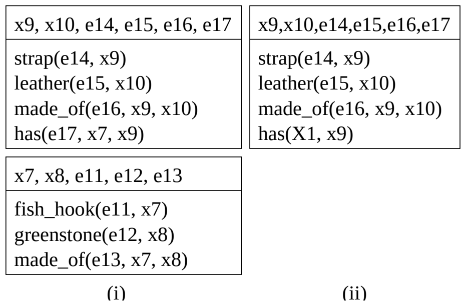

Figure 5: DRS and resolved DRS after Step 3

DRS can be resolved against the model, to model individual $E7$; there is no need for accommodation.

Step 4 is a question, rather than an assertion. The results of this step are shown in Figure 5 (i) and (ii). The presupposed material in the subject DP can again

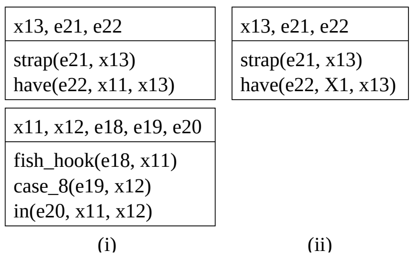

Figure 6: DRS and resolved DRS after Step 4

be resolved, even though it was introduced in separate sentences. The resolved DRS shows that the question to be answered is whether the model individual $X1$ has a strap. The model-checker is able to determine that this is indeed the case, and Te Kaitito therefore responds with `Yes`.

5. Future work

We have several plans for future work. A central goal is to continue expanding the syntactic and lexical coverage of the system. When the coverage is sufficiently expanded, we also plan to adapt the system for use in a second-language-learning system.

References

Barwise, J. and Cooper, R. (1981). Generalized quantifiers and natural language. *Linguistics and Philosophy*, **4**(2), 159–219.

Carroll, J., Copestake, A., Flickinger, D., and Poznanski, V. (1999). An efficient chart generator for (semi-)lexicalist grammars. In *Proceedings of the 7th European Workshop on Natural Language Generation (EWNLG)*, pages 86–95, Toulouse, France.

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

Copestake, A., Lascarides, A., and Flickinger, D. (2001). An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001)*, Toulouse, France.

Creaney, N. (1996). An algorithm for generating quantifiers. In *Proceedings of 8th International Natural Language Generation Workshop*, Sussex, UK.

Dale, R. (1988). *Generating Referring Expressions in a Domain of Objects and Processes*. Ph.D. thesis, Centre for Cognitive Science, University of Edinburgh.

Dale, R., Oberlander, J., Milosavljevic, M., and Knott, A. (1998). Integrating natural language generation and hypertext to produce dynamic documents. *Interacting with Computers*.

Heim, I. (1982). *The semantics of definite and indefinite noun phrases*. Ph.D. thesis, University of Massachusetts. Distributed by Graduate Linguistic Student Association.

Horacek, H. (1996). A new algorithm for generating referential descriptions. In *Proceedings of ECAI*, pages 577–581.

Kamp, H. and Reyle, U. (1993). *From discourse to logic*. Kluwer Academic Publishers, Dordrecht.

Knott, A., Bayard, I., de Jager, S., Smith, L., Moorfield, J., and O'Keefe, R. (2001). Syntax and semantics for sentence processing in english and māori. Manuscript, Dept of Computer Science, University of Otago.

Neumann, G. (1994). *A uniform computational model for natural language parsing and generation*. Ph.D. thesis, University of Saarbrücken.

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

Paris, C., Vander Linden, K., and Lu, S. (2001). Automatic generation of on-line help: a system based on practical issues. In *Proceedings of the 2001 Australasian Natural Language Processing Workshop*, Macquarie University, Sydney.

Piwek, P., Evans, R., and Power, R. (1999). Dialogue as knowledge editing. Technical Report ITRI-99-11, Information Technology Research Institute, University of Brighton.

Pollard, C. and Sag, I. (1994). *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Power, R. (1999). Controlling logical scope in text generation. In *Proceedings of the European Workshop on Natural Language Generation*, Toulouse, France.

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.

Reiter, E. and Dale, R. (2000). *Building natural language generation systems*. Cambridge University Press.

Shieber, S. (1988). A uniform architecture for parsing and generation. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING)*, Budapest.

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

Van Noord, G. (1993). *Reversibility in Natural Language Processing*. Ph.D. thesis, University of Utrecht, The Netherlands.