Corpus-analysis for NLG

Sabine Geldof
Centre for Language Technology
Macquarie University
Sydney, Australia
sabine@ics.mq.edu.au

Abstract

There is a general interest in corpora of human authored texts as a source for acquiring domain knowledge useful for a natural language generation (NLG) system. It is less clear, however, how this can be done in a systematic way. We propose a principled approach towards acquiring domain knowledge through corpus analysis and illustrate its application in the domain of route descriptions. More specifically, we identify different types of knowledge needed in the NLG process and describe a procedure for systematically analyzing a corpus text and for inventorizing these different types of knowledge. We discuss how these procedures fit into a global approach to corpus analysis and into the natural language generation system development cycle.

1 Introduction

1.1 Knowledge requirements for NLG

The process of automatically generating natural language text from non-linguistic input data has been characterized in very general terms by Reiter and Dale (2000) as consisting of three major phases: document planning, micro-planning and surface realization. The simplest architecture to realize this model is a pipeline, where each of the major processes consumes and/or produces an intermediate representation: from a non-textual input over a document plan and a text specification to an output text (see Figure 1). An important argument for considering three phases (rather than only a planning and a realization one) is related to the type of knowledge used in each of these phases.

Document planning consists of transforming a set of non-textual input data into a document plan (i.e. a structured set of messages). At this point only high-level decisions are being made with respect to the contents of the messages and the text structure.

At the other end of the overall NLG process, the surface realization phase receives a detailed text specification ready to be transformed into a final textual form.

As Reiter and Dale note, the phase of document planning – directly related to the non-textual input data – is highly determined by domain specific knowledge, whereas linguistic realization requires mostly linguistic knowledge. However, the gap between the document plan and the text specification requires a number of processing steps involving a combination of linguistic and domain knowledge. These processes are grouped under the term ‘micro-planning’.

Hence it appears that the development of an NLG system, entails the acquisition of domain specific knowledge, especially to assist the first two phases of the process.

Reiter et al. (2000) describe a variety of techniques for acquiring knowledge for use in NLG systems. One specific technique is the analy-

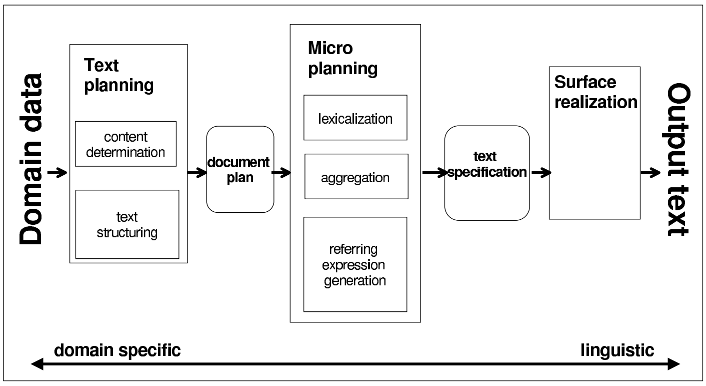
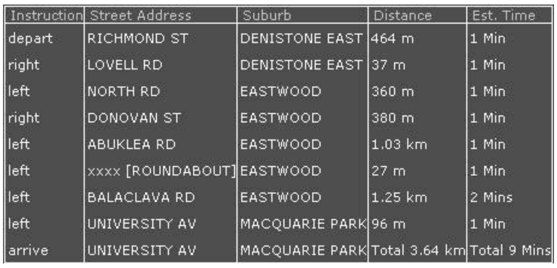


Figure 1: General architecture for NLG system (Reiter and Dale, 2000)

sis of human authored corpora. Reiter and Sripada (2002) note that there is a growing interest in this technique. They describe issues involved in interpreting information extracted from a corpus, warn against the exclusive reliance on corpora for knowledge acquisition and argue for using a good quality corpus rather than a large one. In this paper we investigate which different types of domain-specific knowledge are required (2.1), assuming the general architecture proposed by Reiter and Dale. We propose concrete steps towards analysing a corpus in view of acquiring these types of knowledge (2.2 and 2.3). Finally we describe how the acquired knowledge is integrated in the NLG development cycle (3).



Instruction	Street Address	Suburb	Distance	Est. Time
depart	RICHMOND ST	DENISTONE EAST	464 m	1 Min
right	LOVELL RD	DENISTONE EAST	37 m	1 Min
left	NORTH RD	EASTWOOD	360 m	1 Min
right	DONOVAN ST	EASTWOOD	380 m	1 Min
left	ABUKLEA RD	EASTWOOD	1.03 km	1 Min
left	xxxx [ROUNDABOUT]	EASTWOOD	27 m	1 Min
left	BALACLAVA RD	EASTWOOD	1.25 km	2 Mins
left	UNIVERSITY AV	MACQUARIE PARK	96 m	1 Min
arrive	UNIVERSITY AV	MACQUARIE PARK	Total 3.64 km	Total 9 Mins

Figure 2: An automatically generated route description (obtained from http://www/whereis.com.au)

> Leave the house and drive towards the Midway shops, at the end of the street turn right and then left at the roundabout. Drive along North road and take the third right turn, just after the first hump in the road. Go to the end of that road and then go straight ahead at the roundabout, there's a church on your left. Now go straight along Herring road for quite a way until you hit the main road (Epping Rd), go straight across at the lights and continue on until you get to the next set of lights. Turn right here into the university.

Figure 3: A human generated route description for the route in Figure 2

1.2 The domain of route descriptions

The corpora used in this paper as an illustration belong to the domain of route descriptions: textual explanations of a procedure for reaching a specific destination from a particular point of departure. The issues involved in describing a route are distinct from those concerned with calculating the path sequence underlying the route. Systems providing route descriptions on demand via the web (see for example http://www.whereis.com.au) deliver a route description in both cartographic and textual form. The latter generally consists of a more or less straightforward mapping of the calculated route data (paths to follow and turns to take) to text templates or even to a table, as shown in the example of Figure 2. In the CORAL project, we aim at building an NLG system for producing route descriptions that approximate the naturalness of those produced by humans (see Figure 3) while using input data available in GIS systems that calculate the path. Our approach, described in more detail in Dale et al.(2003), is different from other route description generation systems which assume the availability of rich perceptual (e.g. (Maaß, 1995; Raubal and Winter, 2002)) or semantic information (Pattabhiraman and Cercone, 1990; Moulin and Kettani, 1999). An extensive overview of the field of route description generation systems may be found in the latter paper.

Quite a few studies of route descriptions are based on the analysis of a corpus of human generated directions, often in view of a very specific aspect of this type of discourse. For instance, Klein (1982) illustrated his model for local deixis through examples of route descriptions. Fraczak et al. (1998) used corpus analysis to determine which information in a subway route description is optional or obligatory. Wunderlich and Reinelt (1982) derived from a corpus of route descriptions a model of the interlocutors' interactions. Only Denis (1997) describes a systematic approach to corpus analysis. His aim is to establish measures for qualitative variability among route descriptions. His analysis starts from an assumption of what are essential components in route descriptions. We start from the text, working out what are the building blocks so that an NLG program can approximate it. The corpus used as an illustration in this paper consist of 20 written directions within the urban road network: 9 subjects were asked to describe the route from their homes to the university to a visitor and to a neighbour, and from the university to a fixed, known destination.

2 Corpus analysis approach for NLG

2.1 Domain specific knowledge sources

Assuming the general NLG architecture proposed by Reiter and Dale (2000), we now examine which types of knowledge might be acquired from corpora to support the most domain-sensitive NLG processes.

Document planning decomposes into a process of content determination and one of text structuring. These processes use the following domain specific knowledge sources respectively.

Message types In many NLG systems, messages are created by instantiating message types with data received from another application. Thus the set of possible message types is a knowledge source for the content determination task. A message can generally be characterized as a predicate-argument structure: the predicate asserts a property of an argument or a relationship between arguments and is generally realized as a verb; the argument(s) consist(s) of the main domain entity/ies to which the predicate applies. For example, in our domain, a message might communicate an instruction to follow (predicate) a path (argument). It is not the task of the NLG system to compute this information but rather to determine what configuration of information is appropriate to be communicated as a message.

Text structure patterns The subprocess of text structuring makes explicit the relationship between successive messages in a text. Text structure usually takes the form of a hierarchical structure: a text is composed of different parts, which in turn consist of successive messages in a particular discourse relationship with each other.

In the micro-planning phase, three knowledge sources are used in the refinement of a structured set of messages (the document plan) into a text specification:

Lexemes for message predicates Lexicalization mainly consists in choosing a lexeme to realize each message predicate. Thus, a list of occurring lexemes for each message type predicate is a knowledge source for this process.

RE types for message arguments While the message type specifies which arguments to realize linguistically, the way these domain entities are referred to is still undecided. A list of types of referring expressions for each message type argument constitutes a knowledge source for this process.

Aggregation patterns Finally, human authors tend to combine several messages to form a sentence. Knowledge about which message can be aggregated in which syntactic constructs is needed in order to implement this process.

The following subsections describe major steps in eliciting the above identified knowledge sources.

2.2 Inventorize text planning knowledge

2.2.1 Message types

Analysing a corpus starts by identifying the basic message types occurring in the text. These are to corpus text analysis what objects are to domain modelling: key conceptual elements resulting from segmenting a space into manageable units. Thus the first step in corpus analysis consists in segmenting the text in meaningful units, the next step will be to classify these units into message types. These two analysis processes usually interact, as shown in the procedure described in Figure 5. In this procedure, a key question is what counts as a message. A pragmatic approach is to try to identify the message predicates first (see procedure in Figure 5) and then to further refine the obtained message types on the basis of occurring arguments (see procedure in Figure 6). We will illustrate these procedures with the corpus extract shown in Figure 4.

A first pass segmentation takes the full stops as a boundary (5.1). Then, examining each sentence, we look for re-occurring predicates: what is being communicated? A quite straightforward hypothesis is that route descriptions consist of instructions (hence our hypothesized segmentation criterion, 5.2), more specifically to TURN (in a direction) and to FOLLOW a path. As shown in Figure 4, quite a few sentences occur in between instructions, which we can not label as FOLLOWs or TURNs (5.3). A refinement of our message type list is necessary (5.4). Here the analysis of message arguments (as described in Figure 6) proves useful. A TURN typically takes a *dir* as argument (e.g. ‘left’ in 251-8-2), while a FOLLOW specifies a *path* (‘Burns Bay Rd’ in 251-7-3) (6.1). However, we notice that both TURNs and FOLLOWs can take a *point* (i.e. location where a path ends and where a turn is to be taken) as argument too (6.2b), e.g. ‘after the second one’ in 251-7-2 and ‘right till the end’ in 251-8-1). This leads to the refinement of the TURN and FOLLOW message types into TURN(DIR), TURN(POINT,DIR) and FOLLOW(PATH), FOLLOW(PATH,POINT) respectively. Moreover, both the *point* and *path* arguments occur in non-instructive clauses, as elaborations in between instructions (e.g. 251-10 and 251-7-3). After a few iterations, our analysis results in the following list of message types: TURN(DIR), TURN(POINT,DIR), FOLLOW(PATH), FOLLOW(PATH, POINT), STATE(POINT), STATE(PATH) and the notion of an instruction remains the syntactico-semantic criterion for determining the message boundary.

5.1 Segment the corpus text into sentences;

5.2 List possible message types (on the basis of predicates);

5.3 Identify textual units according to the list of messages identified in the previous step (this may involve taking sentences together or further segmenting them);

5.4 Refine the list of possible message types (on the basis of message arguments (see procedure in Figure 6);

5.5 Go back to step 3 unless

  a. all textual units are labelled according to a coherent list of message types **and**

  b. syntactico-semantic criteria are defined for message boundaries.

Figure 5: Procedure for segmenting text and identifying message predicates

6.1 List for every message type, the set of occurring arguments;

6.2 Check whether each message type selects a disjunct list of arguments,

  a. If yes: done

  b. If no: go to 6.3

6.3 Add a new message sub-type for each frequently occurring combination of predicate and arguments;

Figure 6: Procedure for identifying message-arguments

2.2.2 Text structure patterns

Both the global structure of a text and the relationship among subsequent messages usually af-

Message ID	Text	4.5b	4.2	4.4	7.3	Text
		Segm.	Pred.	Arg.	RE type	Struct.
251-7-1	You'll go over two bridges	DESCR	STATE	*(path)*	*path*-lm	
251-7-2	and after the 2nd one veer to the right	INSTR	TURN	*(point,dir)*	*point*-deic,*dir*	
251-7-3	and you'll be on Burns Bay Rd.	DESCR	STATE	*(path)*	*path*-name	
251-8-1	Stick on this right till the end, which is at Epping Road,	INSTR	FOLLOW	*(path,point)*	*path*-deic, *point*-type-name	
251-8-2	which you'll want to turn left onto	INSTR	TURN	*(dir)*	*dir*-deic	
251-9-1	Epping Road will take you right near the uni,	DESCR	STATE	*(path,point)*	*path*-name, *point*-lm	META
251-9-2	but don't get onto the M2 mistake.	INSTR	TURN	*(dir)*	*dir*-name	META
251-10-1	A couple of kms after the M2 turn off is Herring Road, at the top of a hill.	DESCR	STATE	*(point)*	*point*-name-descr	

Figure 4: Extract from our analysed corpus

fects other decisions along the NLG process (in our domain the message type selection in the first place). It is therefore useful to investigate whether a document typically consists of different parts and how these can be delimited. A well established structure analysis proposed by Wunderlich and Reinelt (1982) distinguishes among the initial route, intermediate routes and the final route. The initial route is defined as the part of the route that can be seen from the departure point. The final route starts where the destination comes within sight of the route taker. We can delimit these parts (DEP, INT, ARR) in our corpus on the basis of message types occurring in these parts of the route descriptions. For instance, nearly all departure messages imply some notion of direction ((TURN(POINT,DIR)), INSTR(POINT)[1], FOLLOW(PATH,DIR)).

The relationship between successive intermediate route messages is quite specific, consisting mainly of an alternation between FOLLOW and TURN messages. However, we already mentioned that elaborations on *point* or *path* arguments may be included to further clarify an element deemed important in the navigation process. These elaborations relate to the instructional messages as satellites to nuclei according to Rhetorical Structure Theory (Mann and Thompson, 1988). There may also be sporadically occurring 'other' messages, which don't follow the just identified rhetorical relationships. In the case of route descriptions, we noted a number of clauses expressing meta information about the route (e.g. 251-9-1) We categorize also negative instructions as META (e.g. 251-9-2).

Thus, our corpus analysis revealed as canonical text structure: three major parts (which select particular message types) and the relationship within the middle part consisting of sequences of nuclei (TURN and FOLLOW messages) with possible elaborations on arguments (points or paths). This knowledge can be formalized as a rewrite grammar.

2.3 Inventorize micro-planning knowledge

2.3.1 Lexemes for predicates

Once the set of message types are determined and the corpus is completely annotated, it is quite straightforward to order the messages by type and to inventorize the lexemes that are representative of each of the message type classes. Table 1 provides an overview of the lexicalization options encountered in our corpus. Lexemes in *italics* face are possible though not encountered in the corpus. The lexicalization patterns also determine how arguments are realized, the ones between brackets are optional. With a few exceptions, lexicalization options are identical across the message subtypes and we have the choice among at least two lexemes for each message (sub)type, e.g. 'follow'

[1] This new message sub-type accounts for a frequently occurring type of instructions: Leave the house (221-1-1), which is neither a TURN, nor a FOLLOW instruction, but rather makes explicit the point of departure in the form of an instruction.

Message type	Sub-type Lexicalization	Sub-type Lexicalization
TURN	TURN(DIR)	TURN(POINT,DIR)
	turn/*veer* $<\text{dir}_{ADVP}>$ (onto/into $<\text{street}_{NP}>$)	(at $< point_{NP} >$) turn/veer $<\text{dir}_{ADVP}>$ (at $< point_{NP} >$)
	take $<\text{dir}_{NP}>$	(at $< point_{NP} >$) take $<\text{dir}_{NP}>$ (at $< point_{NP} >$)
	go $<\text{dir}_{ADVP}>$	(at $< point_{NP} >$) go $<\text{dir}_{ADVP}>$ (at $< point_{NP} >$)
FOLLOW	FOLLOW(PATH)	FOLLOW(PATH,POINT)
	follow $< path_{NP} >$	follow $< path_{NP} >$ until $< point_{NP} >$
	continue along $< path_{NP} >$	continue along $< path_{NP} >$ until $< point_{NP} >$
	stay on $< path_{NP} >$	stay on $< path_{NP} >$until $< point_{NP} >$
	keep going on $< path_{NP} >$	keep going on $< path_{NP} >$ until $< point_{NP} >$
	go down $< path_{NP} >$	go down $< path_{NP} >$ until $< point_{NP} >$
	-	go to the end
	-	go straight until $< point_{NP} >$

Table 1: Overview of lexicalisation options in route descriptions

7.1 Determine the domain entities for which a referring expression generation strategy is needed;

7.2 Determine the analysis parameters that may interact with the RE strategy;

7.3 Identify types of RE for each entity;

7.4 Inventorize REs for the entities identified in 7.1, according to the RE types obtained from 7.3, taking into account the parameters identified in 7.2.

7.5 Go back to 7.3 **unless** the current list of RE types allows to annotate the domain instances.

Figure 7: Procedure for inventorizing referring expressions

and 'continue along' for FOLLOW messages, 'turn' and 'take' for TURN messages.

2.3.2 Types of referring expressions

The procedure for inventorizing types of referring expressions (RE) is similar to the one for lexicalization options but now applies to the arguments of messages instead of the predicates (see Figure 7). In general, the list of arguments identified as part of the message types are the domain entities for which an RE strategy is needed. Thus, in our domain, we inventorized RE types for *path*, *point* and *dir* entities (7.1).

Next, we need to find out whether some aspects of the context in which they appear may have an impact on the choice of a referring expression. We call these analysis parameters. Obviously, the message type is one such parameter. For instance a deictic reference for a *point* occurs only in message type TURN(POINT,DIR). Furthermore, we noticed that in our corpus, an entity might be referred to by a single expression (e.g. 'Burns Bay Rd. for *path* in 251-7-3) or by a combination of expressions (e.g. 'Herring Rd' and 'at the top of a hill' for *point* in 251-10) (7.2). Analysis shows that *points* can be referred to through the name of the intersecting street (*point*-name), the type of intersection (*point*-type, as in 'the end' for a T-junction in 251-8-1), a landmark that occurs at that point (*point*-lm, as 'the uni' in 251-9-1) or a reference to an earlier mention of that point (*point*-deictic, as 'the second one' in 251-7-2). It has to be noted that people use world and perceptual knowledge that is not readily available to the NLG system, such as about the topology of the environment ('at the top of a hill' in 251-10). We have grouped REs for which the underlying information is not available under the common denominator *descriptive* (7.3).

An inventory of the RE types for point shows that mentioning the type of intersection is the only type of RE that occurs across all message types and that it is the most or second most frequently used expression in each message type. It is followed closely (in frequency) by the use of the name of the intersecting street. In contrast to many route descriptions studies (e.g. (Denis, 1997)), which emphasize the importance of landmarks, these don't occur more frequently than the previous types. However, given that they are often used as a single RE, it is still an important RE type.

2.3.3 Aggregation patterns

As for the preceding procedures, we need to determine a set of aggregation patterns for the com-

Aggregation type	**# cases**	**# messages**	**% of total messages**
0	132	133	48
1	57	116	42
2	4	8	3
3	9	18	6
1+	2	2	1
Total	**277 messages**		

Table 2: Occurrence of aggregation patterns in the corpus

bination of messages in sentences and the analysis parameters that might affect their occurrence. Following Reiter and Dale (2000), we distinguish among simple conjunctions (type 1), aggregation with shared constituents or structure (type 2) and syntactic embedding (type 3). Aggregation type 2 occurs only rarely in our corpus. Not surprisingly, since instructive sentences, have no linguistically realized subject (to share). Also the object of a FOLLOW and a TURN message are, by definition, different. Examples of the other two patterns of aggregation do occur frequently: 251-7-1/3 exemplifies the coordination of three messages in one sentence (type 1) and 251-8-1/2 illustrates how a TURN message is syntactically embedded in the preceding FOLLOW message. Furthermore, we labelled the messages which are realized by one sentence (or more) as type 0.

Since we are interested in finding out how messages are combined into sentences, the message type is an obvious analysis parameter, but also the position in the sentence (does this message occur in the first or second position in a conjunction or as main or subordinate clause in the syntactic embedding?). Table 2 shows that most messages are either mapped onto a single sentence (type 0) or simply coordinated (type 1).

As to the relationship between the aggregation type and the message type, further analysis of type 1 aggregation shows that a TURN(POINT-DIR) message is most likely to be coordinated, especially in first position (27/57). In fact, half of these coordinated TURN(POINT,DIR) messages (13/25) are followed by a FOLLOW message. Thus, we might consider TURN(POINT,DIR) *and* FOLLOW (see 251-7-2/3) to be a likely candidate for an aggregation pattern.

3 Integration in NLG development cycle

As described so far, the knowledge sources consist of inventories of constructions occurring in the corpus. They provide a range of choice options for every step in the generation process. We have built and NLG system for route descriptions by combining each of the above knowledge sources with generic NLG processes. The 6 elicited message types (predicates and arguments) constitute the backbone of our system, lexicalisation is kept fairly straightforward and we have implemented an RE strategy for *point* based on the 4 elicited RE types (see details in Dale et al. (2003)). Given input information for a particular route, the system realizes the first choice option for each step in the generation process and, using the mechanism of backtracking, it can realize every possible combination of choice options. The following two fragments (for the route presented in Figure 2) illustrate the range of variation resulting from multiple choice options w.r.t. message type selection and REG.

> Follow Richmond Street for 146m.
> Turn right onto Lovell Road.
> Follow Lovell Road for 37m. until you reach North Road.
> Turn left.

> Follow Richmond Street until you reach the end.
> Turn right. Follow Lovell Road for 37m.
> At the roundabout, turn left onto North Road.

Additional control knowledge is needed to reduce the number of generated alternatives and to determine which constructs to apply in a given situation. Elicitation of control knowledge requires to take into account the context, both linguistic (which message precedes, follows, in which part of the text does this message appear?) and non-linguistic context (i.e. application dependent features of the instances which form the arguments of the messages). This is a multi-dimensional problem, again highly domain dependent, which is the object of our on-going work.

4 Conclusion

As pointed out by Reiter et al.(Reiter et al., 2000), the evaluation of the knowledge acquisition process is very costly and yields only suggestive outcomes. This holds also for our domain, where the effectiveness of descriptions depends on many extra linguistic factors, such as user preferences and physical properties of the environment. At this stage in the development, we consider the capability of generating a range of possible formulations (as cited in Section 3), occurring in a corpus of human generated descriptions to be an important step towards more natural route descriptions (compare these with Figure 2 and 3). We plan however small scale informal evaluations of the system including control knowledge.

While the examples cited are from one corpus, the approach has been used in other corpora belonging to a different sub-domain: directions for navigation by foot on two different campuses. Interestingly, comparing these analyses with the one presented here sheds a light on the differences between these sub-domains and allows us to term these in concrete figures, e.g. about the distribution of message types, RE types and aggregation patterns. Corpus analysis in view of NLG thus contributes to the understanding of the domain.

More importantly, this work is part of an ongoing endeavour to formalize and clearly distinguish NLG knowledge that is generic (hence reusable) from domain specific knowledge which has to be acquired for every new application domain. A systematic approach to corpus analysis contributes to the bottom-up elicitation of these distinctions.

References

R. Dale, S. Geldof, and J.-P. Prost. 2003. Coral: Using natural language generation for navigational assistance. In *Proceedings of the 26th Australasian Computer Science Conference ACSC'03*, Adelaide, South Australia.

M. Denis. 1997. The description of routes: A cognitive approach to the production of spatial discourse. *Current Psychology of Cognition*, 16(4):409–458.

L. Fraczak, G.Lapalme, and M.Zock. 1998. Automatic generation of subway directions: salience gradation as a factor for determining message and form. In *Proceedings of the International Workshop on Natural Language Generation*, pages 58–67, Niagara-on-the-Lake, CA.

W. Klein. 1982. Local deixis in route directions. In R. Jarvella and W. Klein, editors, *Speech, Place and Action. Studies in deixis and related topics.*, pages 161–182. John Wiley & Sons, Ltd.

W. Maaß. 1995. How spatial information connects visual perception and natural language generation in dynamic environments: towards a computational model. Technical Report 116, Universitaet des Saarlandes, FB 14 Informatik IV.

W.C. Mann and S.A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

B. Moulin and D. Kettani. 1999. Route generation and description using the notions of object's influence area and spatial conceptual map. *Spatial Cognition and Computation*, 1:227–259.

T. Pattabhiraman and N. Cercone. 1990. Selection: Salience, relevance and the coupling between domain-level tasks and text planning. In K. McKeown, J.Moore, and S. Nirenburg, editors, *Proceedings of the 5th International Workshop on Natural Langauge Generation*, Dawson, Pennsylvania.

M. Raubal and S. Winter. 2002. Enriching wayfinding instructions with local landmarks. In *GISscience 2002*, Lecture Notes in Computer Science. Springer.

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

E. Reiter and S. Sripada. 2002. Should corpora text be gold standards for nlg? In *Proceedings of the Second International Conference on Natural Language Generation*, pages 97–104, New York, USA.

E. Reiter, R. Robertson, and L. Osman. 2000. Knowledge acquisition for natural language generation. In *Proceedings of the First International Conference on Natural Language Generation*, pages 217–225, Mitzpe Ramon, Israel.

D. Wunderlich and R. Reinelt. 1982. How to get there from here. In R. Jarvella and W. Klein, editors, *Speech, Place and Action. Studies in deixis and related topics.*, pages 182–201. John Wiley & Sons, Ltd.