Planning texts by constraint satisfaction

Richard Power
Information Technology Research Institute
University of Brighton
Lewes Road
Brighton BN2 4GJ, UK
Richard.Power@itri.bton.ac.uk

Abstract

A method is described by which a rhetorical-structure tree can be realized by a text structure made up of sections, paragraphs, sentences, vertical lists, and other textual patterns, with discourse connectives added (in the correct positions) to mark rhetorical relations. We show that text-structuring can be formulated as a Constraint Satisfaction Problem, so that all solutions respecting constraints on text-structure formation and structural compatibility can be efficiently generated. Of the many solutions generated by this method, some are stylistically preferable to others; we show how further constraints can be applied in order to select the best versions. Finally, we discuss some extensions such as the generation of indented text structures.

1 Introduction

Much recent work on language generation (Rosner and Stede, 1992; Hovy, 1993; Mellish et al., 1998) has made use of discourse representations based on Rhetorical Structure Theory (RST) (Mann and Thompson, 1988). Interest has focussed in particular on the problem of building a *rhetorical structure* (RS) which organizes elementary propositions hierarchically by means of RST relations (Marcu, 1996). There has been less attention to a second problem in text planning, that of realizing the RS by a *text structure* (TS), in which the material in the RS is distributed among paragraphs, sentences, vertical lists, etc., perhaps linked up by discourse connectives such as 'since' and 'however'. This task, which we will call *text structuring*, is typically addressed through a micro-planning phase that determines the content of successive sentences. However, documents of realistic complexity require richer TSs including, for example, vertical lists, sub-sections, and clauses separated by semi-colons.

We describe in this paper a text-structuring system that has been developed within ICONOCLAST[1], a project which investigates applications of constraint-based reasoning in Natural Language Generation

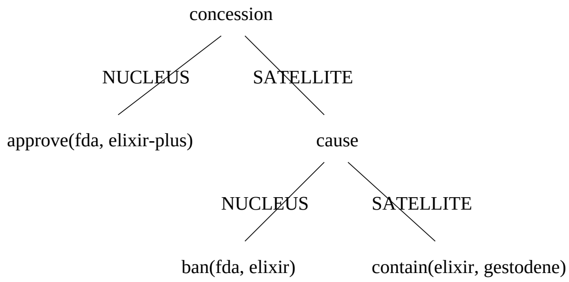

Figure 1: Rhetorical structure

using as subject-matter the domain of medical information leaflets. Following Scott and de Souza (1990), we represent rhetorical structure by graphs like figure 1, in which non-terminal nodes represent RST relations, terminal nodes represent propositions, and linear order is unspecified (for regularity, the nucleus is arbitrarily presented on the left of the satellite). One of many possible TSs realizing this RS is shown in figure 2, an ordered tree in which nodes are labelled with 'text-categories' (Nunberg, 1990); the terminal nodes hold either discourse connectives (which owing to their interaction with text structure have already been selected) or propositions (to be realized in their turn during tactical generation). After passing this TS to the tactical generator, we might obtain the following output[2]:

> The FDA bans Elixir since it contains gestodene; however, the FDA approves ElixirPlus.

Part of the interest of the problem is that RS and TS are not always isomorphic; this will be illustrated later by an alternative TS realizing figure 1 (figure 6b).

Our goal in ICONOCLAST has been to explore the huge variety of ways in which an RS can be conveyed, noting stylistic reasons why one version might be preferred to another, with the eventual aim of providing a system in which the user enjoys fine-grained control over style as well as content. These requirements cannot be met by a text structurer

[1] ICONOCLAST is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) Grant L77102.

[2] The content of the examples is of course fictional.

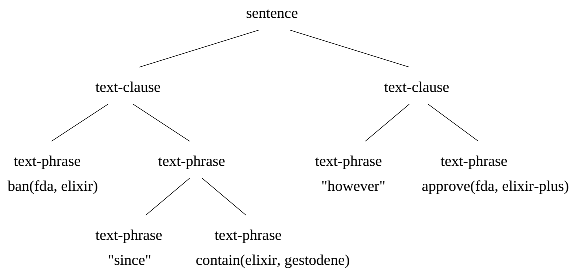

Figure 2: Text structure

that merely returns one or two satisfactory solutions, relying perhaps on a library of schemas. We need a method for enumerating *all* the candidate solutions that can be composed from a given set of text-categories. By a 'candidate' we mean a solution that correctly realizes the RS without violating text-structure formation rules; it may nevertheless be stylistically inept. Having generated a set of candidate text structures, the ICONOCLAST system evaluates them through rules that detect stylistic flaws, and on this basis arranges them in an order of preference. We will discuss stylistic evaluation briefly, but the focus of the paper is the problem of enumerating solutions.

2 Formation rules

A text structure is defined in ICONOCLAST as an ordered tree in which each node has a text-category comprising two features named TEXT-LEVEL and INDENTATION. Values of TEXT-LEVEL are represented by integers in the range $0..L_{Max}$; these may be interpreted in various ways, but we will assume here that $L_{Max} = 4$ and that integers are paired with descriptive labels as follows:

0	text-phrase
1	text-clause
2	text-sentence
3	paragraph
4	section

The meanings of 'section' and 'paragraph' are the usual ones, except that section titles are ignored: a section is simply a sequence of one or more paragraphs. Following Nunberg (1990), 'text-sentence' denotes a unit normally punctuated with a capital letter and a full stop; this is distinguished from the syntactic concept of 'sentence', which depends on syntactic formation rules. Thus the following paragraph consists of three text-sentences which contain, respectively, one, zero, and two syntactic sentences:

> He entered the room. Disaster. The safe was open and the money had gone.

A text-clause is a unit that would normally be punctuated with a semicolon; the text-sentence you are now reading contains two text-clauses, but the second semicolon does not appear because it has been 'absorbed' into the full-stop that marks the whole text-sentence. Within a text-clause, hierarchy is determined by syntax rather than text-structure, so all units within a text-clause are assigned the minimal TEXT-LEVEL of zero.

The purpose of INDENTATION is to allow indented text structures like bulleted lists; the feature takes values in the range 0, 1, 2 ..., where unindented text has INDENTATION = 0,

- a list item has INDENTATION = 1
    - a list item within a list item has INDENTATION = 2

and so forth. To simplify the presentation, we will assume for now that all nodes have INDENTATION = 0, so that text-categories are distinguished only by TEXT-LEVEL.

Informally, a text structure is well-formed if it respects the hierarchy of textual levels, so that sections are composed of paragraphs, paragraphs of text-sentences, and so forth. An example of an ill-formed structure would be one in which a text-sentence contained a paragraph; such a structure can occur only when the paragraph is indented — a possibility we are excluding here. Formally, the text-structure formation rules are as follows:

1. A text structure is an ordered tree in which each node i has a TEXT-LEVEL L_i in the range $0..L_{Max}$.
2. If a node p has a daughter node d, then p must have a TEXT-LEVEL one rank higher than d, unless both nodes have the minimal level 0. In other words, either

    (a) $L_p = L_d + 1$, or
    (b) $L_p = L_d = 0$

    (From this it follows that any nodes that are sisters must have the same level.)
3. All terminal nodes must have the minimal TEXT-LEVEL of 0.

In most applications it would also make sense to set a lower limit on the root node. For instance, we might apply the constraint $L_{Root} \geq 2$ to ensure that the whole text is at least a text-sentence.

3 Compatibility

As well as being a well-formed text structure, a candidate solution must realize a rhetorical structure 'correctly', in a sense that we need to make precise. Roughly, a correct solution should satisfy three conditions:

1. The terminal nodes of the TS should express all the elementary propositions in the RS; they may also contain discourse connectives expressing rhetorical relations in the RS, although for some relations discourse connectives are optional.
2. The TS must respect rules of syntax when it combines propositions and discourse connectives within a text-clause; for instance, a conjunction such as 'but' linking two text-phrases must be coordinated with the second one.
3. The TS must be structurally compatible with the RS.

The first two conditions are straightforward, but what is meant by 'structural compatibility'? We suggest the crucial criterion should be as follows: **any grouping of the elementary propositions in the TS must also occur in the RS**. In other words, the text-structurer is allowed to eliminate groupings, but not to add any. More formally:

- If a node in the TS dominates terminal nodes expressing a set of elementary propositions, there must be a corresponding node in the RS dominating the same set of propositions.
- The converse does not hold: for instance, an RS of the form $R_1(R_2(p_1, p_2), p_3)$ can be realized by a paragraph of three sentences, one for each proposition, even though this TS contains no node dominating the propositions (p_1 and p_2) that are grouped by R_2. However, when this happens, the propositions grouped together in the RS must remain *consecutive* in the TS; solutions in which p_3 comes inbetween p_1 and p_2 are prohibited.

4 Generating solutions

Our procedure for generating candidate solutions is based on a technique for formulating text structuring as a *constraint satisfaction problem* (CSP) (Hentenryck, 1989). In general, a CSP is characterized by the following elements:

- A set of variables $V_1..V_N$.
- For each variable V_i, a finite domain D_i of possible values.
- A set of constraints on the values of the variables. (For integer domains these often use 'greater than' and 'less than'; other domains usually rely on 'equal' or 'unequal'.)

A solution assigns to each variable V_i a value from its domain D_i while respecting all constraints. Depending on the constraints, there may be multiple solutions, or there may be no solution at all.

The difficulty in formulating a configuration task as a CSP is that we usually do not know in advance how many variables the solution will contain. Problems of this kind are sometimes called *dynamic* (Dechter and Dechter, 1988), because the set of relevant variables changes as the search for a solution progresses. The solution in figure 2, for example, has nine TS nodes, each bearing a TEXT-LEVEL variable; different realizations of the same RS might have more nodes, or fewer. However, we have found that all candidate solutions can be generated by assigning four variables (TEXT-LEVEL, INDENTATION, ORDER and CONNECTIVE) to each node of rhetorical structure, so obtaining a *partial description* that determines a unique TS. Intuitively, the idea is that this description should specify a subset of the nodes in the target TS; further nodes are then added, by a deterministic procedure, in order to satisfy the formation rules and accommodate any discourse connectives.

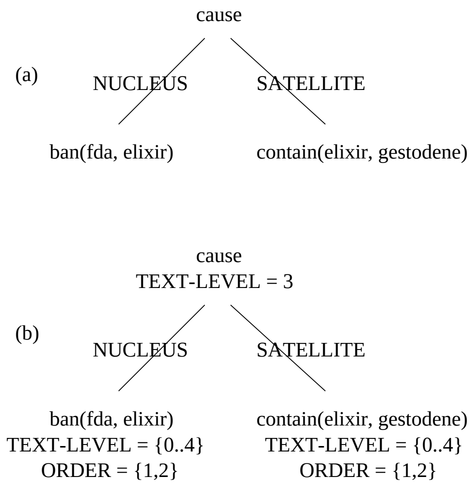

Figure 3: Adding solution variables

As an introduction to this method, we will begin by working through a very simple example. Suppose that our aim is to find all TSs that realize the RS in figure 3a in a paragraph, without using discourse connectives or indentation.

Create solution variables

The first step is to add TEXT-LEVEL and ORDER variables to each RS node. Since ORDER represents the linear position of a text span in relation to its sisters, it can be omitted from the root.

Assign domains

Each variable is assigned a finite domain of possible values (figure 3b). For TEXT-LEVEL variables, the domain is $0..L_{Max}$; for ORDER variables it is $1..N$, where N is the number of sisters. Since we have decided that the whole text should be a paragraph, we can fix the TEXT-LEVEL on the root directly (assigning it the value 3).

Apply constraints
Constraints over the solution variables are now applied. Informally, these are as follows: the root node should have a higher TEXT-LEVEL than its daughters; sister nodes should have the same values for TEXT-LEVEL but different values for ORDER; and since the 'cause' relation is not marked by a discourse connective, its arguments (the two propositions) cannot be realized by text-phrases (the result would be syntactically ill-formed) — in other words, they must have TEXT-LEVEL $\neq 0$. Collectively, these constraints reduce the TEXT-LEVEL domains for the terminal nodes to $\{1,2\}$.

Enumerate solutions
The solutions can now be enumerated by computing all combinations of values that respect the constraints. One example of a solution is shown in figure 4a.

Compute complete text structures
For each solution, a complete TS can be computed by adding any nodes that are required by the text-structure formation rules (figure 4b).

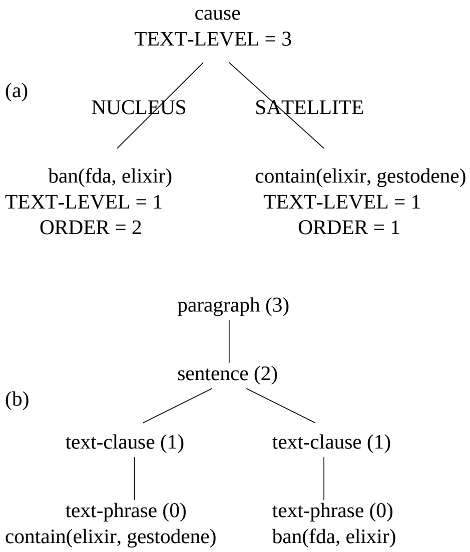

Figure 4: Completing a solution

In this simple case there are just four solutions, since the TEXT-LEVEL and ORDER variables on the nucleus both have the domains $\{1,2\}$, and any setting of these variables fixes the corresponding variables on the satellite. Here are texts that might result from the four solutions (L and O represent TEXT-LEVEL and ORDER; N and S represent nucleus and satellite):

$L_N = 1, O_N = 1, L_S = 1, O_S = 2$
Elixir is banned by the FDA; it contains gestodene.

$L_N = 1, O_N = 2, L_S = 1, O_S = 1$ (figure 4)
Elixir contains gestodene; it is banned by the FDA.

$L_N = 2, O_N = 1, L_S = 2, O_S = 2$
Elixir is banned by the FDA. It contains gestodene.

$L_N = 2, O_N = 2, L_S = 2, O_S = 1$
Elixir contains gestodene. It is banned by the FDA.

The method for including discourse connectives has been described elsewhere (Power et al., 1999). Briefly, the lexical entry for a discourse connective must specify its syntactic category (at present we cover subordinating conjunctions, coordinating conjunctions and conjunctive adverbs) and whether it is realized on the nucleus or the satellite. For example, the relation `cause` can be marked by the subordinating conjunction 'since' (realized on the satellite) or the conjunctive adverb 'consequently' (realized on the nucleus) — among others. The choice of discourse connective strongly constrains the values of TEXT-LEVEL and ORDER for the arguments of the relation. If `cause` is expressed by 'since', the arguments may occur in any order, but they must be text-phrases:

Since Elixir contains gestodene, it is banned by the FDA.

Elixir is banned by the FDA since it contains gestodene.

#Elixir is banned by the FDA; since it contains gestodene.

#Elixir is banned by the FDA. Since it contains gestodene.

If instead `cause` is expressed by 'consequently', the satellite must be placed before the nucleus, and unless the style is very informal the arguments should have TEXT-LEVEL values above text-phrase:

Elixir contains gestodene; consequently, it is banned by the FDA.

Elixir is banned by the FDA. Consequently, it contains gestodene.

#Elixir is banned by the FDA, consequently it contains gestodene.

5 Constraints

We now state the text-structuring constraints precisely, including the feature CONNECTIVE but still omitting INDENTATION. Before applying these constraints, finite domains are assigned to each RS node i:

TEXT-LEVEL $L_i = \{0..L_{Max}\}$

ORDER $O_i = \{1..N\}$ (for N sisters)

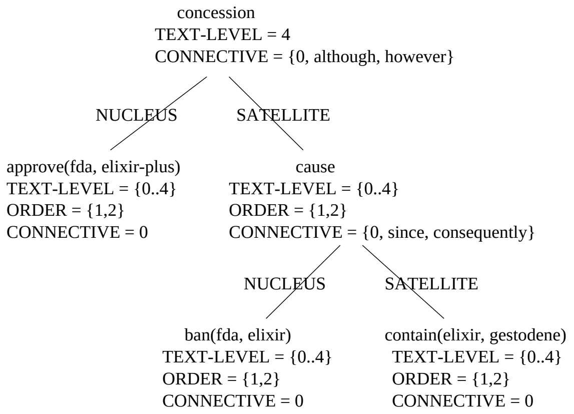

Figure 5: Domain assignments

CONNECTIVE On the node `cause` (figure 3a) $C_i = \{\emptyset, since, consequently\}$; on a proposition node, $C_i = \emptyset$. The value $\emptyset$ represents the option of using no discourse connective.

As an example, possible domain assignments for figure 1 are shown in figure 5. The constraints are as follows:

Root Domination
The TEXT-LEVEL of the root node r must exceed that of any daughter d.

$$L_p > L_d$$

Parental Domination
The TEXT-LEVEL of a parent node p must be equal to or greater than the TEXT-LEVEL of any daughter d.

$$L_p \geq L_d$$

Sister Equality
If nodes a and b are descended from the same parent, they must have the same TEXT-LEVEL.

$$L_a = L_b$$

Sister Order
If nodes a and b are descended from the same parent, they must have different values of ORDER.

$$O_a \neq O_b$$

Argument Order
If C_p is a coordinating conjunction or conjunctive adverb, the argument d (nucleus or satellite) on which the connective will be realised (according to its lexical entry) must have $O_d = 2$.

Subordinating Conjunction Level
If C_p is a subordinating conjunction, any daughter node d (expressing an argument of the relation) must have $L_d = 0$.

Conjunctive Adverb Level
If C_p is a conjunctive adverb, any daughter node d (expressing an argument of the relation) must have $L_d > 0$.

Unmarked Level
If a relation is unmarked ($C_p = \emptyset$) any daughter node d (expressing an argument of the relation) must have $L_d > 0$.

6 Completing the text structure

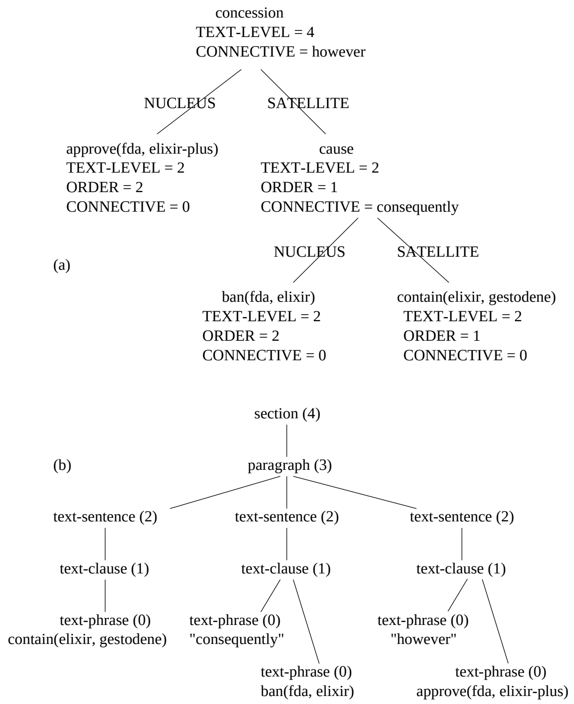

Figure 6: Completing the TS

The algorithm for completing the TS cannot be described fully here, but as an example we comment on how the solution in figure 6a yields the TS in figure 6b.

- If a parent is more than one level above its daughters ($L_p - L_d > 1$), extra nodes are added beneath the parent to bridge the gap — hence the paragraph node in figure 6b.
- If a parent has the same level as its daughters ($L_p = L_d$), the daughters are raised to replace the parent. Thus in figure 6b, the paragraph has *three* sentences, and a rhetorical grouping has been left unrealized in the TS. Of course the reader might infer the intended RS from other evidence (e.g. semantic plausibility).
- If a terminal node i has a level above text-phrase ($L_i > 0$), a chain of nodes is added to bring it 'down to earth' (e.g. the chain below the first text-sentence in figure 6b).

- Discourse connectives are passed down to the text-clause in which they should be realized. This is decided (i) by passing the connective to the appropriate argument (nucleus or satellite), according to its lexical entry, and (ii) by thereafter passing it down to the *first* constituent if the argument is complex (Power et al., 1999).

After tactical generation, we might obtain the following (rather poor) result:

> Elixir contains gestodene. Consequently, it is banned by the FDA. However, the FDA approves ElixirPlus.

7 Style

Having designed a procedure that will generate all text structures meeting minimal standards of correctness, we need to apply further constraints in order to eliminate solutions that are stylistically eccentric — or at least ill-suited to the purpose at hand. In ICONOCLAST, this can be done in two ways:

- If a stylistic defect is regarded as *fatal*, it is excluded by a hard constraint on the solution variables, so that TSs with this defect are never generated.
- If a stylistic defect is regarded as *non-fatal* (i.e. unwelcome but sometimes necessary), it is penalized, by a soft constraint, during a subsequent evaluation phase in which the enumerated solutions are ordered from best to worst.

The user can impose stylistic preferences by switching hard constraints on/off, and also by weighting soft constraints (i.e. determining the importance of non-fatal defects).

We cannot discuss stylistic control in detail here, but we will give one or two examples for each type of constraint.

HARD CONSTRAINTS

Multiple text-clauses: To obtain an informal style without semicolons, sentences containing more than one text-clause can be avoided by imposing the constraint $L_i \neq 1$ on all nodes i.

Nucleus-satellite order: For some rhetorical relations it may be appropriate to fix the linear order of nucleus and satellite; for instance, the satellite of a `background` relation should precede the nucleus. This can be ensured by a constraint $O_S = 1$ on the satellite node S.

SOFT CONSTRAINTS

Rhetorical grouping: Failure to express a rhetorical grouping can be treated as a defect. (This is one reason why the TS in figure 6b is poor.)

Oversimple paragraph: A paragraph containing only one text-sentence can be treated as a defect.

8 Extensions

Our method allows an exhaustive enumeration of solutions, but only within an elementary framework for representing rhetorical and textual structure. We hope to gradually extend this framework to cover many phenomena that are currently excluded:

- Since its input takes the form of a rhetorical structure tree, the text structurer inherits any limitations of RST as a description of rhetorical organization.
- We cover only three types of discourse connective (subordinating conjunction, coordinating conjunction, conjuctive adverb).
- At present there is no treatment of titles.
- There is no treatment of relative clauses, which can be employed for example to realize the `elaboration` relation (Scott and de Souza, 1990):

  > Zovirax, which contains the antiviral agent aciclovir, is a smooth white cream.

- There is no treatment of propositions that are expressed parenthetically.

  > Zovirax, since it is for you only, should never be given to other patients.
  >
  > Zovirax should never be given to other patients (the medicine is for you only).

- We have omitted the colon-expansion pattern (Nunberg, 1990) and some other features influencing punctuation (emphasis, quotation marks, parentheses).
- We have not covered the integration of text with floating items like diagrams, tables, or boxes.

Two extensions that have already been implemented are indentation and centering. We have explained here how indentation is represented in text structure; the relevant constraints will be described elsewhere. Centering has been incorporated by assigning backward and forward centers to all propositions in a completed TS *before* generating the wording; in this way, centering transitions can be evaluated before tactical generation begins, and TSs yielding good continuity of reference can be preferred (Kibble and Power, 1999).

To use our approach in practical applications, one must address the problem that the number of candidate solutions increases exponentially with the complexity of the rhetorical structure — measured, for example, by the number of elementary propositions. In informal trials we find that the number of solutions is roughly 5^{N-1} for an input with N propositions; this means that even for a short passage containing a dozen propositions, the text planner would find about 50 million solutions satisfying the hard constraints. For texts of non-trivial length, there seems no alternative to sacrificing global optimality in the interests of efficiency. One option is to use a statistical optimization method such as a genetic algorithm (Mellish et al., 1998). In ICONOCLAST we have preferred a method of *partial optimization* in which the the text-structuring problem is split into parts, so that at each stage only a manageable part of the total solution is constructed. For instance, when planning a patient information leaflet, the semantic material could first be distributed among sections, then perhaps among paragraphs, thus spawning many small-scale text-structuring problems for which the search spaces would be measured in hundreds rather than billions.

References

A. Dechter and R. Dechter. 1988. Belief maintenance in dynamic constraint networks. In *Proceedings of NCAI-AAAI*. American Association for Artificial Intelligence.

P. Van Hentenryck. 1989. *Constraint Satisfaction in Logic Programming*. MIT Press, Cambridge, Mass.

E. Hovy. 1993. Automatic discourse generation using discourse structure relations. *Artificial Intelligence*, 63:341–386.

R. Kibble and R. Power. 1999. Using centering theory to plan coherent texts. In *Proceedings of the 12th Amsterdam Colloquium*. Institute for Logic, Language and Computation, University of Amsterdam.

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

D. Marcu. 1996. Building up rhetorical structure trees. In *Proceedings of AAAI-96*. American Association for Artificial Intelligence.

C. Mellish, A. Knott, J. Oberlander, and M. O'Donnell. 1998. Experiments using stochastic search for text planning. In *Proceedings of IWNLG-98*, Niagara-on-the-Lake, Canada. Association for Computational Linguistics.

G. Nunberg. 1990. *The Linguistics of Punctuation*. CSLI, Stanford, USA.

R. Power, C. Doran, and D. Scott. 1999. Generating embedded discourse markers from rhetorical structure. In *Proceedings of the European Workshop on Natural Language Generation*, pages 30–38, Toulouse, France.

D. Rosner and M. Stede. 1992. Customizing RST for the automatic production of technical manuals. In R. Dale, C. Mellish, and M. Zock, editors, *Aspects of Automatic Natural Language Generation*, Levico, Italy.

D. Scott and C. de Souza. 1990. Getting the message across in RST-based text generation. In R. Dale, C. Mellish, and M. Zock, editors, *Current Research in Natural Language Generation*. Cognitive Science Series, Academic Press.