An Optimizing Method for Structuring Inferentially Linked Discourse*

Ingrid Zukerman and Richard McConachy
Department of Computer Science
Monash University
Clayton, Victoria 3168, AUSTRALIA
{ingrid,ricky}@bruce.cs.monash.edu.au

Abstract

In recent times, there has been an increase in the number of Natural Language Generation systems that take into consideration a user's inferences. The statements generated by these systems are typically connected by inferential links, which are opportunistic in nature. In this paper, we describe a discourse structuring mechanism which organizes inferentially linked statements as well as statements connected by certain prescriptive links. Our mechanism first extracts relations and constraints from the output of a discourse planner. It then uses this information to build a directed graph whose nodes are rhetorical devices, and whose links are the relations between these devices. The mechanism then applies a search procedure to optimize the traversal through the graph. This process generates an ordered set of linear discourse sequences, where the elements of each sequence are maximally connected. Our mechanism has been implemented as the discourse organization component of a system called WISHFUL which generates concept explanations.

Introduction

Consideration of the inferences an addressee is likely to make from discourse is an essential part of discourse planning. In recent times, there has been an increase in the number of Natural Language Generation (NLG) systems which address the inferences a user is likely to make from the information presented by these systems, e.g., [Joshi et al. 1984; Zukerman 1990; Cawsey 1991; Horacek 1991; Lascarides & Oberlander 1992; Zukerman & McConachy 1993].

A system that addresses a user's possible inferences poses a new set of problems for the discourse structuring component of the system. Consider, for example, the following discourse:

1 `The first step in Bracket Simplification is addition or subtraction.`
2 `For example,` $2(3x+5x) = 2 \times 8x$.
3 *Indeed*, `Bracket Simplification applies to Like Terms.`
4 *In addition*, `as you know, it applies to Numbers.`
5 *However*, `it does not always apply to Algebraic Terms.`
6 `For instance, you cannot add the terms in brackets in` $3(2x+7y)$.

This discourse features inferential relations in lines 2-3, 3-4 and 4-5 (signaled by italicized conjunctions). The sentence in line 3 realizes a generalization from the example in line 2, the sentence in line 4 expands on the information in line 3, and the sentence in line 5 violates an expectation established in line 4.

The two main methods for text organization considered to date are the schema-based approach, e.g., [Weiner 1980; McKeown 1985; Paris 1988], and the goal-based approach, e.g., [Hovy 1988; Moore & Swartout 1989; Cawsey 1990]. Both of these methods are designed to accomplish a single discourse goal. However, inferential relations are opportunistic rather than prescriptive, and therefore cannot be easily cast as contributing to a single communicative goal. Hence, these approaches are ill equipped to cope adequately with inferential links.

In this paper, we present a mechanism which organizes inferentially linked information into maximally connected discourse. This mechanism also copes with prescriptive discourse relations between the intended information and the prerequisite information that is needed to understand the intended information. Our mechanism has been implemented as a component of a system called WISHFUL which generates concept explanations [Zukerman & McConachy 1993].

In the following section, we discuss previous research in discourse structuring. Next, we outline the operation of our discourse planner as background to the description of our discourse structuring mechanism. We then discuss our results and present concluding remarks.

Related Research

The schema based approach was introduced in [Weiner 1980]. It was later formalized in [McKeown 1985] and expanded in [Paris 1988]. This approach consists of compiling rhetorical predicates into a *schema* or template which reflects normal patterns of discourse. Since schemas represent compiled knowledge, they are com-

*This research was supported in part by grant A49030462 from the Australian Research Council.

putationally efficient. However, they do not cope well with the need to exhibit dynamic and adaptive behaviour. This shortcoming is overcome by the goal-based approach.

The two main techniques which represent the goal-based approach are described in [Hovy 1988; Hovy & McCoy 1989] and in [Moore & Swartout 1989]. Both techniques involve converting discourse relations identified in Rhetorical Structure Theory (RST) [Mann & Thompson 1987] into discourse plan operators, and then applying a hierarchical planner [Sacerdoti 1977] to produce a discourse plan. This plan is a tree whose leaves are propositions and whose non-leaf nodes are relations between propositions. Moore's mechanism takes as input a communicative goal, and uses discourse plan operators both to decide what to say and to organize the discourse. Hovy's structurer, on the other hand, is given a set of propositions to be communicated as well as one or more communicative goals. [Hovy & McCoy 1989] later combined Hovy's discourse structurer with *Discourse Focus Trees* proposed in [McCoy & Cheng 1991] in order to enhance the coherence and flexibility of the generated discourse.

The goal-based approach is particularly suitable for situations where a communicative goal may be achieved by whatever means are available, e.g., convincing a user to do something [Moore & Swartout 1989]. However, when the objective is to convey information about a concept, e.g., teach Distributive Law, this approach may omit information that does not fit in the proposed rhetorical structure. For instance, the system described in [Hovy 1988] tries to include as much information as possible in a generated RST tree, but leaves out information that does not fit. The system described in [Cawsey 1990] includes only certain types of information in the discourse operators, and therefore, other relevant information may never be mentioned.

A different approach was taken in [Mann & Moore 1981], where discourse organization is viewed as a problem solving task whose objective is to satisfy some optimality criterion. They implemented a hill-climbing procedure which iteratively selects the best pairwise combination of an available set of *protosentences*. Due to the use of the hill-climbing function, this approach produces locally optimal discourse. In this research, we also view discourse organization as a problem solving task, but we generate discourse which satisfies globally our optimality criterion.

Finally, [Mooney et al. 1991] and [Cerbah 1992] consider the discourse structuring problem at a different level. [Mooney et al. 1991] generate extended discourse by first applying a bottom-up strategy to partition a large number of information items into groups, and then applying a goal-based technique to structure the discourse in each group. [Cerbah 1992] uses global discourse strategies, such as parallel-explanation and concession, to guide the organization of discourse relations in order to generate discourse that achieves a desired overall effect. An interesting avenue of investigation is the adaptation of the mechanism presented in this paper as a component of these systems.

Operation of the Discourse Planner

Our discourse planner receives as input a *concept* to be conveyed to a hearer, e.g., Bracket Simplification; a list of *aspects* that must be conveyed about this concept, e.g., operation and domain; and an *attitude*, which determines a desired level of expertise. It generates a set of *Rhetorical Devices (RDs)*, where an RD is composed of a rhetorical action, such as Assert or Instantiate, applied to a proposition. To this effect, it performs the following steps.

Step 1: WISHFUL first consults a model of the user's beliefs in order to determine which propositions must be presented to convey the given aspects. This step selects for presentation propositions about which the user has misconceptions, and propositions that are believed by the user but not to the extent demanded by the given attitude. Table 1 contains the propositions selected to convey the operation and domain of Bracket Simplification.

p_1: [Bracket-Simplification step-1 +/−]
p_3: [Bracket-Simplification apply-to Like-Terms]
p_5: [Bracket-Simplification ¬(always) apply-to Algebraic-Terms]

Table 1: *Propositions to be Conveyed*

Step 2: Next, WISHFUL applies inference rules in backward reasoning mode in order to generate alternative RDs that can be used to convey each proposition. It then applies inference rules on these RDs in forward reasoning mode in order to conjecture which other propositions are indirectly affected by these RDs. If propositions that are currently believed by the user are adversely affected by inferences from the proposed RDs, they will be added to the set of propositions to be conveyed, e.g., proposition p_4 in Table 2.

Step 3: In this step, the generation process is applied recursively with a revised attitude and new aspects for each of the concepts mentioned in each of the alternative sets of RDs generated in Step 2. This is necessary, since it is possible that the hearer does not understand the concepts mentioned in a particular set of RDs well enough to understand this set. This process generates subordinate sets of RDs, each of which is an alternative way of conveying a concept that was not sufficiently understood by the hearer.

Rhetorical Action	Proposition
Assert *(A)*	p_3: [Bracket-Simplification apply-to Like-Terms]
Mention *(M)*	p_4: [Bracket-Simplification apply-to Numbers]
Assert+Instantiate $2(3x+5x)$ *(A+I)*	p_1: [Bracket-Simplification step-1 +/−]
Negate+Instantiate$^+$ $3(2x+7y)$ *(N+I$^+$)*	p_5: [Bracket-Simplification always apply-to Algebraic-Terms]

Table 2: *The Set of RDs Selected by WISHFUL*

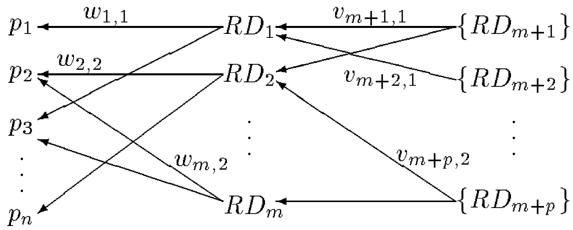

Figure 1: *A 2-layer RD-Graph*

Step 4: For each concept used in each alternative set of RDs, WISHFUL now generates a set of RDs that *evokes* this concept, if the user model indicates that the user and the system do not have a common terminology for it[1]. To ensure that the available discourse options are not constrained unnecessarily, Evocative RDs are generated *before* the organization of the discourse. Further, they are used to generate constraints for the discourse organization process. For instance, consider a situation where the only possible evocation of the concept Like-Terms is "a kind of Algebraic Expression where all the variables are the same." Now, if the organization procedure had been applied before the evocation step, it could have arbitrarily determined that Algebraic-Expressions should be introduced long after Like-Terms. In this case, the resulting discourse would be awkward at best. This situation is avoided by constraining Algebraic-Expressions to appear either before or immediately after Like-Terms. The generation of access referring expressions, on the other hand, must be performed after the organization of the discourse, since decisions regarding pronominalization depend on the structure of the discourse.

Step 5: Owing to the interactions between the inferences from the RDs in each set of RDs generated so far, it is possible that some of the proposed RDs are no longer necessary. In order to remove the redundant RDs, WISHFUL applies an optimization process to each set of RDs. It then selects the set with the least number of RDs among the resulting sets.

The output of the discourse planner is an *RD-Graph*, which is a directed graph that contains the following components: (1) the set of propositions to be conveyed ($p_1, \ldots, p_n$ in Figure 1); (2) the selected set of RDs ($RD_1, \ldots, RD_m, \{RD_{m+1}\}, \ldots, \{RD_{m+p}\}$); (3) the inferential relations between the RDs and the propositions (labelled $w_{i,j}$); and (4) the prescriptive relations between the sets of RDs that generate prerequisite and evocative information and the RDs that are connected to the propositions (labelled $v_{m+k,j}$). The inferential relations are generated in Step 2 above. The weight $w_{i,j}$ contains information about the effect of RD_i on the user's belief in proposition p_j. The prerequisite information is generated in Step 3, and the evocative information in Step 4.

Table 2 contains the set of RDs generated by WISHFUL for the input in Table 1. The rhetorical action

[1] Evocation pertains to the first time a concept is mentioned in a piece of discourse, as opposed to *access*, which pertains to subsequent references to this concept [Webber 1983].

Mention indicates that the user is familiar with the proposition in question. **Instantiate$^+$** stands for an Instantiation annotated with a short explanation, such as that in line 6 in the sample discourse in the Introduction. The algebraic expressions $2(3x + 5x)$ and $3(2x+7y)$ in the Instantiations are the objects on which the corresponding propositions are instantiated.

Operation of the Discourse Structurer

Our discourse structuring mechanism generates an optimal ordering of the RDs in the RD-Graph generated by the previous steps of WISHFUL. Our optimality criterion is *maximum connectivity*, which stipulates that the generated discourse should include the strongest possible relations between the RDs in the graph.

Our procedure first uses the relations in the RD-Graph to derive constraints and relations that affect the order of the generated RDs. The constraints are strict injunctions regarding the relative ordering of these RDs, while the relations are suggestions regarding the manner in which the RDs should be connected. These constraints and relations are then represented as a *Constraint-Relation Graph*, which is a directed graph whose nodes are RDs and whose links are relations and constraints. Finally, we apply a search procedure which finds the optimal traversal through the graph, i.e., the traversal which uses the strongest links and violates no constraints.

Extracting Constraints and Relations

The constraints extracted by our mechanism are BEFORE and IMMEDIATELY-AFTER. They are obtained directly from the prescriptive links in the RD-Graph (the links in the right-hand layer of the graph in Figure 1) by applying the following rule:
If $\exists$ a link between $\{RD_{m+k}\}$ and RD_j ($v_{m+k,j} \neq 0$)
Then BEFORE($\{RD_{m+k}\}$,RD_j) or
IMMEDIATELY–AFTER($\{RD_{m+k}\}$,RD_j).

These constraints stipulate that a set of RDs that is used to evoke or explain a concept must be presented in the discourse either at any time before this concept is presented or immediately after it.

The relations extracted by our mechanism are CAUSE, REALIZE, ADD and VIOLATE. The first three relations represent corroborating information, where the causal relation is the strongest, and the additive relation the weakest. The fourth relation represents conflicting information. In order to derive these relations, the system first obtains *support* and *soundness* information from the weights $w_{i,j}$ of the inferential links in the RD-Graph (the links in the left-hand layer of the graph in Figure 1).

Support indicates whether an inference from an RD supports or contradicts a proposition. Inferences that support a proposition are *positive (+)*, while inferences that contradict it are *negative (−)*.

Soundness indicates the level of soundness of an inference from an RD. We distinguish between three types of positive inferences based on the soundness of the inference rules that yield these inferences: *sound (s)*, *acceptable (a)* and *unacceptable (u)*. Negative inferences are not affected by this distinction, since the

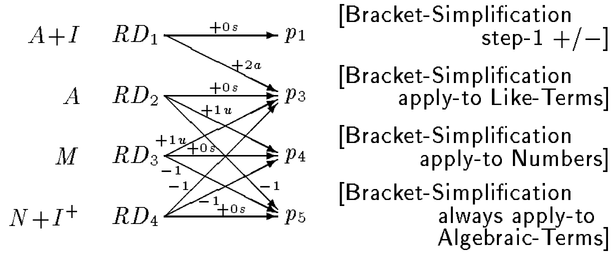

Figure 2: *RD-Graph for the Selected Set of RDs*

manner in which they are addressed is not influenced by their soundness.

Sound inferences are logically sound, e.g., a specialization from a positive statement or a generalization from a negative statement.

Acceptable inferences are sensible, and their results hold often, e.g., a generalization from an instance to a class, a generalization from a positive statement, or a specialization from a negative statement.

Unacceptable inferences hold only occasionally, and hence should not be reinforced, e.g., inferences based on the superficial similarity of two items.

In addition, the discourse structurer requires **directness** information, which conveys the length of the inference chain which infers a proposition from an RD. A directness of level 0 corresponds to a direct inference, level 1 corresponds to inferences drawn from the application of one inference rule such as generalization or specialization, level 2 corresponds to the combination of two inference rules, etc. Directness reflects the intentionality of the discourse, since direct inferences are usually the ones intended by the speaker. Hence, an RD that conveys a proposition by means of a direct inference always has a positive support for this proposition[2]. Directness information is obtained directly from the inference rules used by the system.

Figure 2 depicts support, soundness and directness information for the RD-Graph which corresponds to the set of RDs in Table 2. For instance, the label $+2a$ represents an acceptable inference of positive support and directness 2.

The relations between the RDs are derived from these factors by means of the procedure *Get-Inferential-Relations*. For each proposition, the algorithm builds a set of binary relations of the form $Rel(RD_i, DirRD)$. Each binary relation contains one RD that conveys this proposition directly ($DirRD$), and another that affects it indirectly (RD_i). As stated above, the possible values of Rel considered by our mechanism are: VIOLATE, CAUSE, REALIZE and ADD. The relation VIOLATE is obtained first from the RDs that affect a proposition indirectly with a negative support, i.e., $DirRD$ is at odds with each of these RDs. The remaining RDs, which have a positive support, corroborate $DirRD$. They are divided into those from which the proposition is derived by means of a sound inference, those from which the proposition is inferred by an acceptable inference, and those which yield the proposition through an unacceptable inference. These RDs are related to $DirRD$ by means of the relations CAUSE, REALIZE and ADD, respectively. Table 3 contains the binary relations generated by our algorithm for the RD-Graph in Figure 2.

Procedure **Get-Inferential-Relations**(*RD-Graph*)
For each proposition $p \in$ *RD-Graph* do:

1. $DirRD \leftarrow$ the RD from which p is inferred directly.
2. $IndRD \leftarrow$ the RDs from which p is inferred indirectly.
3. If $IndRD \neq \emptyset$ and $DirRD \neq \emptyset$ Then
   {VIOLATE(RD_i,$DirRD$)| RD_i affects p with a *negative* inference}
   {CAUSE(RD_i,$DirRD$)| RD_i affects p with a *sound* and *positive* inference}
   {REALIZE(RD_i,$DirRD$)| RD_i affects p with an *acceptable* and *positive* inference}
   {ADD(RD_i,$DirRD$)| RD_i affects p with an *unacceptable* and *positive* inference}

Building the Constraint-Relation Graph

After the ordering constraints and relations have been extracted from the RD-Graph, they are combined in order to generate the Constraint-Relation Graph used in the next step of the discourse organization process. This is done by iteratively adding each constraint and relation to a graph that starts off empty, without disrupting the links built previously. In order to support a numerical optimization process, the links in the Constraint-Relation Graph are assigned weights. Constraints (BEFORE and IMMEDIATELY-AFTER) are assigned a weight of ∞, since constraints must never be violated. Relations are assigned weights according to their support and soundness as follows: CAUSE 4, REALIZE 2, VIOLATE 2 and ADD 1. Figure 3 illustrates the Constraint-Relation Graph built from the relations in Table 3.

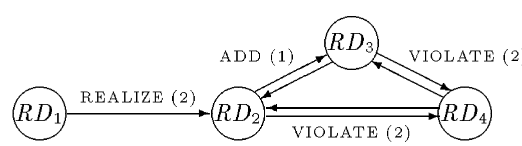

Figure 3: *The Constraint-Relation Graph*

Generating the Optimal Traversal

The procedure for generating the optimal traversal of the Constraint-Relation Graph consists of three stages: (1) *path extraction*, (2) *filtering*, and (3) *optimization*.

The **path extraction** stage generates all the *terminal paths* starting from each node in the Constraint-Relation Graph, where a terminal path is one that continues until a dead-end is reached. For instance, node RD_1 in Figure 3 has two alternative terminal paths: (1) RD_1 – REALIZE – RD_2 – VIOLATE – RD_4 – VIOLATE – RD_3, and (2) RD_1 – REALIZE – RD_2 – ADD – RD_3 – VIOLATE – RD_4.

[2] The Negation of proposition p has a positive support for the intended proposition $\neg p$.

	p_1	p_3	p_4	p_5
DirRD	RD_1	RD_2	RD_3	RD_4
IndRD $(-)$	–	RD_4	RD_4	$\{RD_2, RD_3\}$
IndRD $(+s)$	–	–	–	–
IndRD $(+a)$	–	RD_1	–	–
IndRD $(+u)$	–	RD_3	RD_2	–
Relation		REALIZE(RD_1,RD_2) ADD(RD_3,RD_2) VIOLATE(RD_4,RD_2)	ADD(RD_2,RD_3) VIOLATE(RD_4,RD_3)	VIOLATE(RD_2,RD_4) VIOLATE(RD_3,RD_4)

Table 3: *Relations Extracted from the RD-Graph*

The **filtering** stage deletes *redundant* and *irregular* paths, where a path is redundant if there exists another path which subsumes it; and a path from node RD_i to node RD_j is irregular if it contains consecutive VIOLATE links, and there exists another path from node RD_i to node RD_j that is composed of positive links only. For example, the path RD_2 – ADD – RD_3 – VIOLATE – RD_4 is redundant, since it is subsumed by path (2) above. The first path above is irregular, since there is a positive link, namely ADD, between RD_2 and RD_3. The deletion of redundant paths cuts down the search, and the deletion of irregular paths prevents the generation of sentences of the form "RD_2, but RD_4. However RD_3" if a sentence of the form "RD_2 and RD_3. However RD_4" is possible.

The **optimization** stage consists of applying algorithm A* [Nilsson 1980], where the goal is to select an ordered set of terminal paths which covers all the nodes in the Constraint-Relation Graph, so that the sum of the weights of the links in these paths is maximal. The operators for expanding a node in the search graph are defined as follows:

Operator O_i traces the terminal path $path_i$ through the Constraint-Relation Graph, and removes from the graph the nodes along the traced route and the links incident upon these nodes. The application of O_i generates discourse that connects the RDs in $path_i$.

After the application of an operator, the problem state consists of (1) the terminal paths removed so far from the Constraint-Relation Graph, and (2) the remaining part(s) of the Constraint-Relation Graph. The remaining parts of the graph must then be processed similarly until the graph is empty.

A* uses the evaluation function $f(n) = g(n) + h(n)$ for each node n in the search graph, and terminates the search at the node with the highest value of f. In order to satisfy the admissibility conditions of A*, g and h are set to the following values:

$$g = \sum_{path \epsilon Paths} \sum_{\{RD_i, RD_j\} \epsilon path} weight(link_{RD_i,RD_j})$$

$$h = \sum_{RD_i \epsilon \{CRG-Paths\}} Weight_{RD_i} - \min_{RD_i \epsilon \{CRG-Paths\}} Weight_{RD_i}$$

where *Paths* are the paths removed so far from the Constraint-Relation Graph *CRG*; $weight(link_{RD_i,RD_j})$ is the weight of the link which connects RD_i and RD_j; and $Weight_{RD_i}$ is the maximum of the weights of the links incident on RD_i.

The h function estimates the best possible outcome based on the remaining parts of the Constraint-Relation Graph. This outcome corresponds to the discourse that would result if the strongest link incident on each node could be used in the terminal path that covers the remaining graph. The weakest among these links is subtracted from the h function, since $n-1$ links are needed to connect n nodes.

The result of applying this procedure to the Constraint-Relation Graph in Figure 3 is the ordered discourse sequence RD_1 – REALIZE – RD_2 – ADD – RD_3 – VIOLATE – RD_4 which has a total weight of $2 + 1 + 2 = 5$. This sequence yields the following output, which corresponds to the sample text in the Introduction.

Assert+Instantiate$\{2(3x + 5x)\}$
[Bracket-Simplification step-1 +/−]
REALIZE
Assert [Bracket-Simplification apply-to Like-Terms]
ADD
Mention [Bracket-Simplification apply-to Numbers]
VIOLATE
Negate+Instantiate$^+\{3(2x + 7y)\}$
[Bracket-Simplification always apply-to Algebraic-Terms]

Handling Constraints

Our mechanism also handles the constraints BEFORE and IMMEDIATELY-AFTER. Recall that these constraints involve a set of RDs which evokes or explains a singleton RD, e.g., BEFORE($\{RD_{m+k}\}$,RD_j). The discourse structurer extracts constraints and relations from the set $\{RD_{m+k}\}$ and builds a Constraint-Relation Graph as explained above. This graph is subordinate to the node RD_j in the main graph, and it is linked to RD_j by a BEFORE/IMMEDIATELY-AFTER hyper-link. The optimization process is applied separately to this graph, resulting in a connected sequence of RDs for the set $\{RD_{m+k}\}$.

This sequence is treated as a single entity when the terminal paths are built for the main graph. When the BEFORE link is followed, this sequence yields an introductory segment that appears at some point before RD_j. Alternatively, when the IMMEDIATELY-AFTER link is followed, it yields a subordinate clause. In this case, if the subordinate graph contains only a few RDs, the main path may continue after the subordinate clause. For example, "Bracket Simplification applies to Like Terms, *which are Algebraic Terms such as* $3(2x + 5x)$. In addition, it applies to Numbers." However, if the subordinate graph is large, the terminal path must stop immediately after it in order to avoid an unwieldy tangential discussion.

Results

As stated above, the mechanism described in this paper is part of a system for the generation of concept explanations. This system is implemented in Sun Common Lisp on a SPARCstation 2. The example discussed throughout this paper takes approximately 4 CPU seconds to reach the stage shown in Table 2, and an additional second to produce the final ordered output sequence of rhetorical devices and relations. Since the discourse organization problem is exponential, the mechanism is slowed down by larger input patterns with many inter-relationships which produce large, highly connected Constraint-Relation Graphs. For example, it takes about twenty seconds to structure one sample input of twenty RDs.

The preliminary testing of our mechanism has been performed in the domains of algebra (14 examples) and zoology (7 examples). Our mechanism was also informally evaluated by showing hand-generated English renditions of its output to staff and tutors in the Department of Computer Science at Monash University. The general opinion of the interviewed staff was that the text was logically constructed. In addition, a comparison of the output of our mechanism with texts in prescribed textbooks showed that this output follows the layout of published instructional material.

Conclusion

We have offered a discourse structuring mechanism that organizes inferentially linked rhetorical devices as well as rhetorical devices linked by prerequisite relations. Our mechanism extracts ordering constraints and inferential relations from the output of a discourse planner, and optimizes the ordering of the generated rhetorical devices based on the principle of maximum connectivity. The output of this mechanism captures sufficient rhetorical features to support continuous discourse.

References

Cawsey, A. 1990. Generating Explanatory Discourse. In Dale, R., Mellish, C., and Zock, M. eds. *Current Research in Natural Language Generation.* Academic Press.

Cawsey, A. 1991. Using Plausible Inference Rules in Description Planning. In Proceedings of the Fifth Conference of the European Chapter of the ACL.

Cerbah, F. 1992. Generating Causal Explanations: From Qualitative Models to Natural Language Texts. In Proceedings of the Tenth European Conference on Artificial Intelligence, 490-494, Vienna, Austria.

Horacek, H. 1991. Exploiting Conversational Implicature for Generating Concise Explanations. In Proceedings of the Meeting of the European Association for Computational Linguistics.

Hovy, E.H. 1988. Planning Coherent Multisentential Text. In Proceedings of the Twenty-Sixth Annual Meeting of the Association for Computational Linguistics, 163-169, Buffalo, New York.

Hovy, E.H.; and McCoy, K.F. 1989. Focusing Your RST: A Step Toward Generating Coherent Multisentential Text. In Proceedings of the Eleventh Annual Meeting of the Cognitive Science Society, 667-674, Ann Arbor, Michigan.

Joshi, A.; Webber, B.L.; and Weischedel, R.M. 1984. Living Up to Expectations: Computing Expert Responses. In Proceedings of the National Conference on Artificial Intelligence, 169-175, Austin, Texas.

Lascarides, A. and Oberlander, J. 1992. Abducing Temporal Discourse. In Dale, R., Hovy, E., Rösner, D., and Stock, O. eds. *Aspects of Automated Language Generation.* Springer-Verlag, Berlin, Heidelberg.

Mann, W.C.; and Moore, J.A. 1981. Computer Generation of Multiparagraph English Text. *American Journal of Computational Linguistics* 7(1): 17-29.

Mann, W.C.; and Thompson, S.A. 1987. Rhetorical Structure Theory: A Theory of Text Organization, Technical Report No. ISI/RS-87-190. Information Sciences Institute, Los Angeles, California.

McKeown, K. 1985. Discourse Strategies for Generating Natural Language Text. *Artificial Intelligence* 27(1): 1-41.

McCoy, K.F.; and Cheng, J. 1991. Focus of Attention: Constraining What Can Be Said Next. In Paris, C.L., Swartout, W.R., and Mann, W.C. eds. *Natural Language Generation in Artificial Intelligence and Computational Linguistics.* Dordrecht, The Netherlands: Kluwer Academic Publishers.

Mooney, D.J.; Carberry, S.; and McCoy, K.F. 1991. Capturing High-level Structure of Naturally Occurring, Extended Explanations Using Bottom-up Strategies. *Computational Intelligence* 7(4): 334-356.

Moore, J.D.; and Swartout, W.R. 1989. A Reactive Approach to Explanation. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 1504-1510, Detroit, Michigan.

Nilsson, N. 1980. *Principles of Artificial Intelligence.* Palo Alto, California: Tioga Publishing Company.

Paris, C.L. 1988. Tailoring Object Descriptions to a User's Level of Expertise. *Computational Linguistics* 14(3): 64-78.

Sacerdoti, E. 1977. *A Structure for Plans and Behaviour.* New York, New York: Elsevier, North-Holland Inc.

Webber, B.L. 1983. So What Can We Talk About Now? In Brady, M., and Berwick, R.C. eds. *Computational Models of Discourse.* MIT Press.

Weiner, J. 1980. Blah, A System Which Explains Its Reasoning. *Artificial Intelligence* 15: 19-48.

Zukerman, I. 1990. A Predictive Approach for the Generation of Rhetorical Devices. *Computational Intelligence* 6(1): 25-40.

Zukerman, I.; and McConachy, R.S. 1993. Generating Concise Discourse that Addresses a User's Inferences. Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, Chambery, France. Forthcoming.