

# Review of the Literature on Aggregation in Natural Language Generation\*

Eric J. S. Joanis

Department of Computer Science

University of Toronto

joanis@cs.toronto.edu

September 26, 1999

## **Abstract**

In this paper, we review the significant body of research on aggregation, especially in the past decade. The linguistic phenomena labelled *aggregation* are distinguished and classified and their use in Natural Language Generation is analyzed. Several systems using aggregation are described and used as examples of what can be done, and their architectures are compared. Finally, a chronology of significant contributions to aggregation is provided.

---

\*The author wishes to acknowledge the support of the National Sciences and Engineering Research Council of Canada and the Bell Canada University Labs.

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Types of Aggregation</b>	<b>4</b>
<b>3</b>	<b>Systems Using Aggregation</b>	<b>7</b>
<b>4</b>	<b>Details of Each Type of Aggregation</b>	<b>10</b>
4.1	Embedding . . . . .	10
4.1.1	Sample Embedding Operators . . . . .	12
4.1.2	Paraphrasing . . . . .	13
4.2	Paratactic Aggregation . . . . .	14
4.3	Lexical Aggregation . . . . .	17
4.3.1	Bounded Lexical Aggregation . . . . .	18
4.3.2	Unbounded Lexical Aggregation . . . . .	18
<b>5</b>	<b>System Architectures for Aggregation</b>	<b>20</b>
5.1	Independent Sentence Planning . . . . .	21
5.2	Opportunistic Text Planning . . . . .	22
5.3	Discourse Organization Module . . . . .	23
5.4	Revision-based generation . . . . .	23
<b>6</b>	<b>Conclusions</b>	<b>24</b>
<b>A</b>	<b>A Chronology of Contributions</b>	<b>25</b>

## 1 Introduction

In the past few years, aggregation has become a popular research topic in Natural Language Generation (NLG). However, the researchers' approaches differ vastly and are often not compatible. Also, the term *aggregation* has been applied to several more or less related problems. The objectives of this paper are to review the work that has been done so far on aggregation, to sort it out, and to classify the different types of aggregation.

Aggregation is generally viewed as the task of taking several separate input elements and combining them in a more complex structure, which is intended to be more concise and more fluent. This task is or should be performed in some way or another in virtually every multi-sentence NLG system, especially systems with highly formulaic input (*e.g.*, text generated from database records as in MAGIC [McKeown *et al.*, 1997]). It is often considered to be a text polishing step (*e.g.*, HealthDoc [DiMarco *et al.*, 1997], REVISOR [Callaway and Lester, 1997], PLANDOC [Shaw, 1995]), but in some systems it is an integral part of text planning (*e.g.*, STREAK [Robin, 1994a], ILEX [Cheng *et al.*, 1997]). It can also be used as a summarization tool (*e.g.*, STREAK and PLANDOC [McKeown *et al.*, 1995]). The best definition of aggregation, provided by Cheng *et al.* [1997], encompasses well what other researchers consider this task to be:

*Functioning as one or a set of processes acting on some intermediate text structures in text planning, aggregation decides which pieces of structures can be combined together to be realized as complex sentences later on so that a concise and cohesive text can be generated while the meaning of the text is kept almost the same as that without aggregation.*

Several linguistic phenomena fall under the label of *aggregation*, but different authors classify them differently. Section 2 compares the classifications proposed by various authors and extracts one unified classification.

Section 3 gives a short description of several NLG systems using aggregation. Section 4 describes the types of aggregation in detail and explores how they are used in NLG projects. The task of paraphrasing, although not directly related to aggregation, works hand in hand with aggregation in some systems, so it is described in Section 4.1.2. Section 5 describes types of architectures found in NLG systems performing aggregation. Finally Section 6 opens up on areas of further research and conclusions and Appendix A gives a chronology of significant contributions to aggregation.

## 2 Types of Aggregation

Several papers provide a classification of the types of aggregation [Dalianis, 1996a, Shaw and McKeown, 1997, Reiter and Dale, 1997, Cheng *et al.*, 1997]. These classifications are based on the linguistic phenomena involved. Wilkinson [1995] and Reape and Mellish [1999] provide an orthogonal classification based on the locus of aggregation, that is, based on the representation used to perform the aggregation. Since this paper focuses on the linguistic phenomena, this second classification will not be described.

The four authors giving a linguistic classification of aggregation each describe four types of aggregation, but these types differ from author to author. Table 1 shows the correspondence between these classifications and the one retained in this paper, which is described below in further detail.

		Dalianis		Shaw & McKeown	Reiter & Dale	Cheng <i>et al.</i>
Embedding				Hypotactic	Embedding	Embedding
						Hypotactic
Paratactic		Syntactic		Paratactic	Simple conjunction	Paratactic
					Ellipsis	
Lexical	BL	Lexical	BL	Referential	Set formation	Lexical
	UL		UL	Semantic		

Table 1: Types of Aggregation

Of all the classifications, those of Cheng *et al.* [1997] and Dalianis [1996a] are the most clearly stated, and the one retained is a combination of those two. Cheng *et al.* [1997] distinguish between embedding and hypotactic aggregation by restricting hypotactic aggregation to dependent clauses linked by subordinating conjunctions, but this case is not relevant to aggregation, so the distinction is not useful. Hence in this review, following most other authors, the terms *embedding* and *hypotactic aggregation* will be used interchangeably.

Dalianis’s distinction between bounded (BL) and unbounded (UL) lexical aggregation, omitted by Cheng *et al.* [1997] because their work does not focus on lexical aggregation, is kept here because the two problems are in fact very different [Dalianis, 1996a, Dalianis and Hovy, 1996]. Dalianis adds two further types, elision (the suppression of information which can be inferred) and referential aggregation (mostly the use of anaphora), but these problems are not really related to aggregation, so they are omitted here.

Reiter and Dale [1997] distinguish between simple conjunction, *i.e.*, the usage of conjunction in paratactic aggregation, and ellipsis, *i.e.*, the removal of the resulting redundant elements, but other authors consider these two steps as integral parts of paratactic aggregation, so this distinction is not retained.

A further classification is provided by Horacek [1992], who calls aggregation *grouping*. “Content-based grouping” is related to UL aggregation, “structurally motivated, purely propositional grouping” is a specialized type of paratactic aggregation, and “structurally motivated grouping involving quantifications” deals with quantifiers and finite sets, somewhat like BL aggregation. Fehrer and Horacek [1997] use the hearer’s knowledge to omit steps in a logical proof, implementing a form of aggregation related to what Dalianis [1996a] calls elision.

The classification retained in this paper distinguishes four types of aggregation: embedding, paratactic aggregation, bounded lexical (BL) aggre-

gation, and unbounded lexical (UL) aggregation, defined as follows:

**Embedding** or **Hypotactic aggregation** inserts the contents of a proposition as a sub-constituent in a main proposition.

**Paratactic aggregation** conjoins propositions and potentially elides redundant constituents.

**Lexical aggregation** replaces a set of related items by a single term which conveys the meaning of the set.

**Bounded lexical (BL) aggregation** is used with well-defined, fixed-sized sets. No meaning is lost in the aggregation.

**Unbounded lexical (UL) aggregation** is used when the set is potentially unlimited and some inference is required to choose the aggregated term. Some meaning is lost in the aggregation.

Table 2 gives an informal description of the operation performed by each type of aggregation so as to appeal to the reader’s intuition, along with short illustrative examples.

Type	Operation
Example	
Embedding	Insert a modifier
<i>There is a car on the street; the car is red → There is a red car on the street</i>	
Paratactic	Join with a conjunction
<i>John likes Mary; John likes Suzy → John likes Mary and Suzy</i>	
BL	Replace a bounded set
<i>Monday, Tuesday, Wednesday, Thursday, and Friday → Every week-day</i>	
UL	Replace an open set
<i>Microsoft, Corel, and Inprise → Software companies</i>	

Table 2: Examples of Aggregation

### 3 Systems Using Aggregation

Many NLG systems use aggregation at some stage to produce their output. These systems will be quoted frequently in the rest of the paper, so it is worth describing each one individually.

PLANDoc [McKeown *et al.*, 1994, Shaw, 1995, McKeown *et al.*, 1995, Kukich *et al.*, 1997] produces a one to two page summary of an engineer’s interaction with Bellcore’s long-term telephone network planning tool, the LEIS<sup>1</sup>-PLAN system. The input is a trace of the engineer’s interaction with the system, and the output describes the scenarios and refinements considered by the engineer, in a format suitable for presentation to management or for inclusion in training material.

MAGIC (Multimedia Abstract Generation for Intensive Care) [McKeown *et al.*, 1997, Dalal *et al.*, 1996a, Dalal *et al.*, 1996b] “generates a multimedia briefing that integrates speech, text, and animated graphics to provide an update on patient status” to various hospital personnel immediately after a surgical operation. The input includes the patient’s medical profile and online data collected during the operation. The output includes labelled animated graphics synchronized with spoken text. Key concerns for the generated spoken text include conciseness: the care-givers are pressured for time, so they need a short text, but which must be unambiguous; and media-specific tailoring: the generated text should be appropriate for spoken language.

CASPER (Clause Aggregation in Sentence Planner) [Shaw, 1998a, Shaw, 1998b] is the sentence planner used in PLANDoc and MAGIC. It performs hypotactic and paratactic aggregation, paraphrasing and sentence boundary determination.

---

<sup>1</sup>LEIS is a registered trademark of Bell Communications Research, Piscataway, NJ.

FLOWDoc [Passonneau *et al.*, 1996, Kukich *et al.*, 1997] automatically documents work-flow diagrams used in business re-engineering. The input comes from SHOWBiz, a tool used to present and manipulate work-flow diagrams, and the output is a few natural language sentences summarizing the work flow. It uses ontological generalization (UL aggregation) to summarize information about similar activities and entities.

ZEDDoc [Kukich *et al.*, 1997, Passonneau *et al.*, 1997] summarizes traffic for advertisements on the WWW. The input data are logs of WWW page hits. This project is primarily an exercise in software re-use, with large parts of this system ported from PLANDoc and FLOWDoc.

STREAK (Surface Text Revisor Expressing Additional Knowledge) [Robin, 1994a, Robin, 1994b, McKeown *et al.*, 1995, Robin and McKeown, 1996] generates concise summaries of basketball game results. The input consists of a set of obligatory facts and a set of floating, or optional, facts to be included opportunistically, *i.e.*, if they can be expressed concisely. The output is a single complex sentence comparable to a human-written newswire summary. It is intended to contain as much information as possible as concisely as possible.

REVISOR [Callaway and Lester, 1997] implements revision-based generation. The test corpus used is multi-sentential botanical explanations, but the approach is intended to apply to explanations in general. The input is a series of facts or propositions about the topic of interest, and the output is a fluent text presenting all this information. The novelty of this system is the performing of revisions on an abstract discourse plan.

ILEX (Intelligent Labelling Explorer) [Cheng *et al.*, 1997, Cheng, 1998] is a virtual museum system. It presents natural-language artefact de-

scriptions in a hypertext setting and adapts the descriptions on the basis of what the user has already seen in the current virtual tour. The system input is a network of facts about the artefacts in the virtual museum, called the Content Potential. The output descriptions contain the basic facts about the artefact, as well as additional information opportunistically inserted by use of embedding (see Section 5.2 for further details on opportunistic generation).

HealthDoc [DiMarco *et al.*, 1997, Hovy and Wanner, 1996, Wanner and Hovy, 1996] is a generic architecture for the generation of user-tailored natural-language documents based on the generate-and-repair paradigm. The test domain is customized medical patient education material. The input is a master document marked to indicate which sections of the text should be included as a function of user characteristics. The output of the selection engine is a potentially disfluent text which includes all the information that will be contained in the final text. The repair phase is then responsible for making the text fluent.

*PROVERB* [Fiedler and Huang, 1995, Huang and Fiedler, 1996, Huang and Fiedler, 1997, Fiedler, 1996, Fiedler, 1997, Fiedler, 1998] is a proof verbalizer. It takes natural deduction style machine generated logic proofs and renders them in natural language. Its microplanner, *PROMI* [Fiedler, 1996], uses aggregation to improve fluency and remove redundancies in the verbalization.

VINST (VIsual and Natural language Specification Tool) [Dalianis, 1995b] generates natural language paraphrases of visual and formal specifications in the domain of telephone communications. The input is a formula in a first-order-logic language extended with time, LOXY, and the output is a natural language paraphrase of the formula.

Delphi Tool [Dalianis, 1995a] paraphrases formal specifications in the Delphi language, again in the domain of telephone communications. It uses aggregation to remove redundancies found in the formal language.

Spokesman [Meteer and Shaked, 1988, Meteer, 1991, Meteer, 1992] is Meteer's implementation of her Text Structure abstract linguistic representation. It is used as an interface between the surface realizer Mumble-86 and a variety of applications, such as the Main Street simulation program, the Semi-Automated Force Project and the AirLand Battle Management Project. The system is meant to fill what Meteer calls the generation gap, or the need for text planning to take into account, and therefore have access to, linguistic realization considerations.

## 4 Details of Each Type of Aggregation

In this section, we describe each type of aggregation in detail and explain how it is used in various NLG systems.

### 4.1 Embedding

Embedding consists of taking a proposition and inserting its contents as a sub-constituent into another proposition. The embedded proposition is shifted to a simpler syntactic form and modifies one of the constituents of the main proposition [Cheng *et al.*, 1997]. For example, *There is a car on the street; the car is red*  $\rightarrow$  *There is a red car on the street.*

Embedding is the most interesting and powerful type of aggregation. It can be used to express multiple facts about an entity concisely, and it is an effective tool in opportunistic generation (see Section 5.2). Cheng [1998] uses it to generate what she calls the non-referring part of a referring expression, *i.e.*, the part which is not required to identify the entity, but

which supplies additional information about it. It can be applied to the ELABORATION RST (Rhetorical Structure Theory [Hovy, 1990b]) relation [Scott and de Souza, 1990]. In general, it requires that the two propositions have some entities in common [Shaw, 1998a]. Using the Generalized Upper Model (GUM), Cheng *et al.* [1997] apply embedding to some Being&Having configurations such as Ownership (*e.g.*, *A man with black hair*), Identity (*e.g.*, *My friend Jack*), and Property-Ascription (*e.g.*, *a young student*); and some Doing&Happening configurations such as Nondirected-Doing (*e.g.*, *The walking man*) and Nondirected-Happening (*e.g.*, *The dying man*).

Three types of embedding are given by Scott and de Souza [1990]: nominal, adjectival and adverbial. Each type has a number of possible realizations with different complexities. A nominal can be embedded as a noun or an appositive phrase [Scott and de Souza, 1990]. For example, given *King made this jewel; King is a Scottish designer*, the second proposition can be embedded as a noun: *The Scottish designer King made this jewel* [Cheng *et al.*, 1997], or as an appositive phrase: *King, a Scottish designer, made this jewel*. An adjectival can be embedded as an adjective, a prepositional phrase or a relative clause. For example, given *A man bought the picture; the man had blond hair*, the second proposition can be embedded as an adjective: *A blond man bought the picture*, as a prepositional phrase: *A man with blond hair bought the picture*, or as a relative clause: *A man who had blond hair bought the picture* [Scott and de Souza, 1990]. An adverbial can be realized as an adverb or a prepositional phrase. For example, given *Paula danced with Peter; she was willing*, the second proposition can be embedded as an adverb: *Paula danced with Peter willingly*, or as a prepositional phrase: *Paula danced with Peter with willingness* [Scott and de Souza, 1990]. In the examples quoted above, the simpler form (given first in the three examples) is also the more readable one. This illustrates a heuristic proposed by Scott and de Souza [1990]: when many forms of embedding are possible for the same propositions, prefer the simplest one.

Shaw [1998a] empirically confirms that this heuristic corresponds to human authoring preferences. There are, however, exceptions to this rule. Notably, physicians prefer using prepositional phrases to identify the disease of a patient, such as in *a patient with diabetes*, even when a simpler adjectival form like *a diabetic patient* exists, because the prepositional form is more expressive, allowing the embedding of disease qualifiers, *e.g.*, *a patient with severe type-1 diabetes*, and because not all diseases have an adjectival form, *e.g.*, *peptic ulcers* [Shaw, 1998a]. This leads to another heuristic for choosing the embedded form: if related concepts are to be embedded in the same proposition, prefer the simplest form which can accommodate them together [Shaw, 1998a]. Taking the idea even further, STREAK [Robin, 1994a] searches the revision space for the optimal form, defined to be the form which embeds the most facts in a reasonably complex sentence.

#### 4.1.1 Sample Embedding Operators

For illustrative purposes, this section presents a few embedding operators from different systems. These operators are Object Embedding, used in *PROVERB* [Huang and Fiedler, 1996], the embedding of Being&Having configurations in ILEX [Cheng *et al.*, 1997], and adjunctization and nominalization in STREAK [Robin, 1994a, McKeown *et al.*, 1995].

Object Embedding is used in *PROVERB* [Huang and Fiedler, 1996] when the same mathematical object is found in non-identical parallel structures. For example,  $Set(F) \wedge Subset(F, G)$  (*F is a set; F is a subset of G*) can be aggregated to  $Subset(Set(F), G)$ , *the set F is a subset of G*.

As mentioned in the previous section, ILEX [Cheng *et al.*, 1997] will embed some Being&Having GUM configurations. For example an ownership relation can be expressed as a post-modifier: *A man came to the party; he had black hair*  $\rightarrow$  *A man with black hair came to the party*. Conversely, a role played by an entity can become a pre-modifier: *He announced the beginning of the conference; he is the chairman*  $\rightarrow$  *As the chairman, he announced the*

*beginning of the conference.*

In STREAK [Robin, 1994a, McKeown *et al.*, 1995], the adjunctization rule mixes paraphrasing and embedding. It applies to sentences which use a support verb (a verb where the meaning is primarily carried by the object rather than the verb itself), *e.g.*, *Charles Barkley scored 42 points Sunday*. To embed the historical fact *He tied a season high*, the object *42 points* is adjunctized to an instrument position and the support verb is replaced by a full verb: *Charles Barkley tied a season high with 42 points Sunday* [Robin, 1994a].

The converse rule to adjunctization is nominalization [Robin, 1994a, McKeown *et al.*, 1995]. Again, paraphrasing is used to enable the embedding of further information. In this case, a full verb is replaced by a support verb and a nominal phrase which can accept new modifiers. For example, the full verb *defeated* in *The Phoenix Suns defeated the Dallas Mavericks 123–97* can be nominalized to include historical information: *The Phoenix Suns handed the Dallas Mavericks their 13th straight home defeat 123–97* [Robin, 1994a].

#### 4.1.2 Paraphrasing

As illustrated by the two last examples, paraphrasing is a linguistic tool which works hand in hand with embedding. It consists of choosing a different syntactic realization for a particular concept. Given a main proposition and a candidate proposition for embedding, it might not be possible or stylistically desirable to perform the operation with the main proposition as is. For example, *Mrs. Jones is a diabetic patient; she has severe type-2 diabetes* should not be aggregated to *Mrs. Jones is a severe type-2 diabetic patient*. Instead, it is better style to paraphrase the main proposition, recasting the adjective *diabetic* to the PP *with diabetes*, which can accept the desired modifier: *Mrs. Jones is a patient with severe type-2 diabetes* [Shaw, 1998a].

Paraphrasing can be used to deal with style considerations [Scott and de Souza, 1990], as illustrated by the example above. It can also be used to explore the space of possible realizations for given contents, and choose an optimal (or near-optimal) realization. This approach is used in STREAK [Robin, 1994a, McKeown *et al.*, 1995] and, to a lesser extent, in *PROVERB* [Huang and Fiedler, 1996, Huang and Fiedler, 1997].

STREAK [Robin, 1994a] uses embedding and paraphrasing extensively. In fact, it does not distinguish between paraphrasing and embedding. Instead, it has a bank of revision operators, which includes pure embedding rules, pure paraphrasing rules and a number of mixed rules (*e.g.*, the adjunctization and nominalization rules described in Section 4.1.1). The revision space is explored, and all possible combinations of the rules are tried to find a realization which includes the most facts together and is expressed as concisely as possible (see Section 5.2 for more details).

*PROVERB* [Huang and Fiedler, 1996] makes a restricted use of paraphrasing. For a given concept, various realizations are explored primarily to avoid “building inexpressible text structures”. For example, given the logic formula  $derive(\text{para}(C1, C2), B)$ , where  $B$  is some hypothetical conclusion,  $\text{para}(C1, C2)$  could be expressed as the clause *C1 and C2 are parallel* or the nominal phrase *the parallelism of C1 and C2*, amongst other choices. However, if the *derive* relationship is verbalized as *B, since A*, then only the clausal realization can be used: *B, since C1 and C2 are parallel*, but not *\*B, since the parallelism of C1 and C2*. Conversely, the verbalization *Because of A, B* would require the nominal realization.

## 4.2 Paratactic Aggregation

Paratactic aggregation consists of conjoining two or more propositions with the help of a coordinating conjunction. Redundant information can be elided so that the conjunction can be found at any depth in the sentence structure. For example, *John has a car; John drives to school* can be aggregated to

*John has a car and drives to school.* At a deeper nesting level, *John walks with Mary*; *John walks with Jane* can be aggregated to *John walks with Mary and Jane*.

Early research on paratactic aggregation [Scott and de Souza, 1990, Dalianis and Hovy, 1993, Fiedler and Huang, 1995, Fiedler, 1996] focuses on finding specific rules identifying syntactic structures on which it applies. Scott and de Souza give a set of psycholinguistics-based heuristics for applying paratactic aggregation on an RST tree. Dalianis’s Predicate and Subject Grouping rules apply parataxis when the predicates or subjects, respectively, of two propositions are identical. *PROVERB* [Fiedler, 1996] includes a domain-specific version of the same rules for mathematical proofs.

McKeown’s group uses a more general approach, implemented in *CASPER* (Clause Aggregation in Sentence Planner) [Shaw, 1998a, Shaw, 1998b], the sentence planner used in *MAGIC* [McKeown *et al.*, 1997] and *PLANDOC* [McKeown *et al.*, 1994, Shaw, 1995].<sup>2</sup> *CASPER*’s coordination, or parataxis, algorithm is divided in four steps [Shaw, 1998b]:

1. order and group propositions by similarities,
2. identify recurring elements,
3. determine sentence boundaries,
4. elide appropriate recurring elements.

This approach is both powerful and efficient [Shaw, 1998b]. Step 2 only marks recurring elements, letting step 4, done by the lexical chooser, perform all elision. This way, *CASPER* has the flexibility to apply parataxis at any syntactic level and to keep the appropriate instance (first or last) of a recurring element. Going back to the walking example, step 2 would mark

---

<sup>2</sup>*FLOWDOC* and *ZEDDOC* are successors of *PLANDOC* and reuse much of its code [Kukich *et al.*, 1997], so they might use *CASPER* as well. However, Shaw [1998a] omits *FLOWDOC* and *ZEDDOC*, which suggests they don’t.

*John*, *walks*, and *with* as recurring elements, and step 4 would determine that the second occurrences are to be elided: *John walks with Mary and ∅ ∅ ∅ Jane*.<sup>3</sup> In contrast, the first instance of the recurring time modifier will be elided in *Al re-stocked coffee on Monday; Al removed rotten milk on Monday*, which will become *Al re-stocked coffee ∅ and ∅ removed rotten milk on Monday*. As these two examples illustrate, CASPER has a set of rules determining which instance of a recurring element can be deleted, if at all, so that a sentence like *\*∅ walks with Mary and John walks ∅ Jane* could not be produced [Shaw, 1998b].

The efficiency of the approach is primarily due to the fact that CASPER does all aggregation before lexicalizing. The result might not be the best that can be achieved, since some factors are not considered, but the whole system runs much faster [Shaw, 1998a]. In contrast, STREAK [Robin, 1994a] fully lexicalizes every alternative structure considered, and as a result it can take very long to plan some complex sentences [Shaw, 1998a, Callaway and Lester, 1997].

An important issue, raised by Scott and de Souza [1990], mentioned only briefly by Shaw [1998b], is determining when it is possible to apply parataxis. Of the RST relations, Scott and de Souza [1990] propose that only the multi-nuclear ones allow parataxis: SEQUENCE, CONTRAST, LIST, and ALTERNATIVE. Shaw [1998b] identifies three types of coordination, *segregatory*, *combinatory* and *rhetorical*. Segregatory coordination is equivalent to the coordination of clauses (*e.g.*, *John likes Mary and Jane* corresponds to *John likes Mary; John likes Jane*). Combinatory conjunction introduces a different meaning (*e.g.*, *Mary and Jane are sisters* is not equivalent to *Mary is a sister; Jane is a sister*). Rhetorical conjunction is used when a coordinating conjunction marks a rhetorical relation (*e.g.*, *The boat arrived and [then] unloaded*). Shaw [1998b] restricts the application of paratactic aggregation to segregatory coordination. It is not clear, however, how

---

<sup>3</sup>Following the standard convention, ∅ is used to mark ellipsis.

CASPER [Shaw, 1998b] prevents the aggregation of input propositions that would create undesired combinatory or rhetorical coordination. For example, given *Mary is a sister; Jane is a sister* as input, how would the system know not to apply parataxis? Moreover, parataxis can introduce ambiguities, for example, *Mary has a pen; Jane has a pen* would be aggregated to *Mary and Jane have a pen*, which has two interpretations: *Mary and Jane have a pen together* or *Mary and Jane each have a pen* [Dalianis, 1996c].

### 4.3 Lexical Aggregation

Lexical aggregation has been studied significantly less than syntactic (hypotactic and paratactic) aggregation. Bounded lexical aggregation (BL) is fairly simple and Dalianis and Hovy [1996] deal with most of the issues. On the other hand, unbounded lexical aggregation (UL) requires involved domain-specific inferences, making a general approach difficult to develop. Dalianis and Hovy [1996] provide a description of the problem, but no adequate solution.

Lexical aggregation corresponds to three types of aggregation mentioned by Wilkinson [1995] and Reape and Mellish [1999]. These authors base their classification on the locus of aggregation (*i.e.*, where it is performed in an NLG system) rather than the linguistic result (*i.e.*, how the aggregated text differs from the non-aggregated one). As a result, they distinguish between conceptual, semantic and lexical aggregation: conceptual aggregation is performed on the conceptual representation, which is assumed to be language independent; semantic aggregation, on the language dependent semantic representation; lexical aggregation, on the lexical or surface representation. Since we are concerned with the linguistic phenomena in this paper, those distinctions are not retained. However, although BL aggregation can be performed at all three levels, UL aggregation usually requires deeper information, and would be performed either at the conceptual or the semantic level.

### 4.3.1 Bounded Lexical Aggregation

BL aggregation is a relatively simple problem. It consists of identifying the occurrence of a list containing most or all the elements of a bounded set, and replacing it by the name of the set, with any exceptions marked by appropriate cue words such as *except* and *all... but...* [Dalianis and Hovy, 1996]. Little inference is required; the system only needs to know about the set and its elements. Example sets on which BL can be performed are the days of the week (or the work week), the months of the year, and the ten digits [Dalianis and Hovy, 1996]. It also applies to limb pairs, *e.g.*, *the right arm and the left arm* aggregates to *both arms*.

Although the BL algorithm is straightforward, a good knowledge representation is required to store and efficiently retrieve the bounded sets known to the system [Dalianis and Hovy, 1996]. Alternatively, the system could logically infer what items belong to a set. These questions have not been addressed in any research that I am aware of.

### 4.3.2 Unbounded Lexical Aggregation

UL aggregation is probably the hardest problem that falls under the *aggregation* label. Simply defined, it replaces a set of related elements by a subsuming one. For example, *Microsoft, Netscape and Corel* can be aggregated to *software companies* (Graeme Hirst, personal communication). The classical example is about fighting: *John hit Peter; Peter hit John back* can be aggregated to *John and Peter fought* [Hovy, 1990a, Horacek, 1992, Dalianis and Hovy, 1996]. Dalianis [1996a] also aggregates *selling tomatoes, purchasing cars, vending CDs, buying stocks, selling dollars, buying houses, and holding a garage sale* to *doing business*. These few examples show the range of summarization that UL can achieve. In the software companies example, the specific companies are abstracted away, but no other information is lost. In the doing-business example, more information is lost, since the activities aggregated together are of a wide variety. The choice of ag-

gregated text could also be altered to keep more or less information: *three software companies* retains the count, whereas *companies* would even lose the type of companies included. The aggregation process needs to be context sensitive to produce good results. The simple term *companies* might be sufficient when the software companies are contrasted to universities, but a more precise term would be required if they were contrasted with other software companies, *e.g.*, smaller ones or ones targeting a different market.

Few systems implement UL aggregation. Besides the work in FLOWDOC [Passonneau *et al.*, 1996] described below, Dalianis and Hovy [1996] have a toy implementation where each UL rule must be defined explicitly. PAULINE [Hovy, 1990a] has a fancier algorithm which can choose to include details, interpretations or both. The interpretations can be the result of UL aggregation or some other inference mechanism. For example, when describing a fight between *John* and *Peter*, an unaggregated text describing each insulting, pushing, and hitting action gives the details of the event, whereas the aggregated text *John and Peter fought* only provides an interpretation. Hovy quotes this fighting example, a clear case of UL aggregation, but also some election-result interpretations which do not fit within the scope of aggregation.

In order to perform UL aggregation, a grouping of concepts must be available. Hovy [1990a] and Dalianis and Hovy [1996] use hand-crafted grouping rules. A more general approach, implemented in FLOWDOC [Passonneau *et al.*, 1996] and mentioned by Shaw and McKeown [1997], is to use ontological subsumption. A general ontology like WordNet, the Generalized Upper Model or some other generic conceptual hierarchy, augmented or accompanied by a domain-specific ontology, could be used to identify the generalization which best subsumes a given set of concepts. For FLOWDOC [Passonneau *et al.*, 1996], the ontology comes from SHOWBIZ, the process re-engineering tool providing FLOWDOC with its input.

The ontological subsumption algorithm used in FLOWDOC [Passonneau

*et al.*, 1996] takes a set of objects  $C_O$  and attempts to find generalizations which optimize the following two competing criteria:

- **Coverage**, the number of objects which are subsumed by the generalization;
- **Specificity**, the semantic distance between each covered object and the generalization.

If necessary to get full coverage, a set of several subsuming terms can be used, but the algorithm gives a *verbosity* penalty for using more than one term. A *heterogeneity* penalty is also given for having terms that are too deep in the ontology. For example, the generic term *Document* would be favored over the too-specific term *Draft document in SGML format*. This choice is consistent with the preference for basic-level words described by Dale and Reiter [1996] and originally by Rosch *et al.* [1976].

This subsumption algorithm is reminiscent of Minimal Message Length (MML) techniques because it attempts to balance the length of the message (roughly analogous to the depth in the ontology) and the probability of recovering the original message (the unaggregated text) (Ingrid Zuckerman, personal communication; see [Oliver and Hand, 1994] for an introduction to MML). However, Eduard Hovy (personal communication) believes that this similarity is metaphorical and that the mathematics of MML do not apply to ontological generalization or UL aggregation.

## 5 System Architectures for Aggregation

The systems which implement aggregation are designed in a number of different ways. Their architectures can be classified under four categories, on the basis of where aggregation is performed. Again, this classification differs from that of Wilkinson [1995] and Reape and Mellish [1999], which focuses primarily on the representation on which aggregation is performed. Here, we are concerned with the role aggregation plays in the overall architecture and how it interacts with other NLG tasks. The architecture types we retain

are:

- Independent sentence-planning module,
- Opportunistic text planning,
- Discourse organization module,
- Revision-based generation.

The first and second types are radically opposite approaches, and in fact serve different generation goals. In particular, opportunistic text planning is mostly appropriate for summarization systems. The third type combines techniques from the first two and the fourth type describes a technique that can be used in conjunction with the other types.

### 5.1 Independent Sentence Planning

Independent sentence planning treats aggregation as a microplanning sub-task which happens after text planning and before surface realization. When it runs, the full contents have been selected and the rhetorical structure of the text is determined. The module attempts to improve the text quality by performing aggregation on propositions which are either adjacent or close enough to be brought together, but the interaction with other NLG tasks is minimal. This architecture is used in *PROVERB* [Huang and Fiedler, 1996], *VINST* [Dalianis, 1995b] and *Delphi Tool* [Dalianis, 1995a].

*HealthDoc* [Hovy and Wanner, 1996, Wanner and Hovy, 1996, DiMarco *et al.*, 1997] also falls under this category, even though aggregation is performed along with other types of text repair tasks, all working together in the blackboard architecture. It could be argued, however, that the flexibility of the blackboard architecture should place *HealthDoc* in the category of systems with a discourse organization module described below.

## 5.2 Opportunistic Text Planning

Opportunistic text planning views aggregation as a tool used by text planning or content selection. It is used to generate concise summaries of some input data. The first planning task is to determine what information is essential, and what information is optional. All essential information is included in the text plan, but then aggregation, especially embedding, is used to determine which optional facts should be included. If an optional fact can be expressed concisely, usually by being embedded in an existing part of the draft text structure, it is included; otherwise it is left out. This approach is implemented in STREAK [Robin, 1994a, McKeown *et al.*, 1995] and ILEX [Cheng *et al.*, 1997].

In ILEX [Cheng *et al.*, 1997], a system which describes objects in a virtual museum, an initial text structure is built from the basic facts, with open slots where additional modifiers can be inserted. The bank of facts available about the object is then searched for facts which can be embedded in these slots, until the most informative text structure is built or all information is included. For example, if a sentence is to be generated to describe the designer of a jewel and we have the basic fact *designer(jewel, King)*, the additional facts *Scottish(King)* and *work-place(King, London)* can be embedded to yield *This jewel was designed by the Scottish designer Jessie King who worked in London* [Cheng *et al.*, 1997].

In STREAK [Robin, 1994a], the base expression is the result of a basketball game, and the additional information comes from historical data about the teams and players involved. The operators used to find the most informative text integrate paraphrasing and embedding, as well as paratactic aggregation, allowing for a search over the space of combinations of facts and their possible realizations [Robin, 1994a, McKeown *et al.*, 1995]. However, to avoid generating excessively complex sentences, a limit of 45 words and 10 levels of syntactic embedding is imposed, respecting the limits observed in the corpus of human-written game summaries [McKeown *et al.*, 1995].

### 5.3 Discourse Organization Module

This approach combines ideas from the first two approaches. It assumes that the content selection is done as a first independent step and that all facts passed on must be included, but then uses the technique of opportunistic planning to build the text structure, yielding a concise and flowing text. Meteer's Text Structure implementation, Spokesman [Meteer, 1991] and the revision-based system REVISOR [Callaway and Lester, 1997] use this approach. McKeown's group also uses this approach in several NLG systems, including PLANDOC [Shaw, 1995], CASPER [Shaw, 1998a, Shaw, 1998b], MAGIC [McKeown *et al.*, 1997], and FLOWDOC and ZED-DOC [Kukich *et al.*, 1997]. Judging by the success of the applications using this approach, it is one of the best ways to perform aggregation when content determination can be done up-front without regard to realization issues. Indeed, this approach gives the application a high degree of flexibility.

### 5.4 Revision-based generation

Revision-based generation does not contrast with the former types; rather it is a particular feature of some systems in the other categories. Revision-based generation is based on human writing analysis described by Hayes and Flower [1986], which concludes that writing is usually done in three heavily interwoven (and recursively nested) phases: planning, sentence generation, revision. Since humans generate multiple drafts before producing a final text, Callaway and Lester [1997] and Cheng *et al.* [1997] argue that it is reasonable (and presumably desirable) for NLG systems to do the same. Systems like REVISOR [Callaway and Lester, 1997], HealthDoc [DiMarco *et al.*, 1997], ILEX [Cheng *et al.*, 1997], STREAK [Robin, 1994a] use this architecture because they all generate an initial draft and then apply various revision rules to modify it. In the case of REVISOR and HealthDoc, the revisions reorganize the text without adding new contents, but in ILEX and STREAK the revision operators opportunistically add new information.

A unique feature of REVISOR is that it performs the revisions on an abstract discourse plan, that is, a discourse plan stripped of all semantic and syntactic features which are not required to determine what revisions are applicable. The lighter-weight data structure allows for a much faster search through the revision space, especially since intermediate abstract plans are not verbalized; only the final plan is [Callaway and Lester, 1997]. In contrast, STREAK fully realizes each candidate and intermediate plan. However, the revision operators used in REVISOR are limited to simple cases of hypotactic and paratactic aggregation, so it is not clear how much Callaway and Lester’s approach can scale to a revision space with the complexity of STREAK’s.

## 6 Conclusions

With all the research that has been done about aggregation so far, many problems have been solved. STREAK, ILEX and CASPER provide a solid basis for dealing with embedding. CASPER provides the most comprehensive algorithm for paratactic aggregation. FLOWDOC provides a promising approach for UL aggregation. CASPER and REVISOR propose approaches that can be used to perform aggregation efficiently.

However, many problems remain open. FLOWDOC’s ontological generalization algorithm has only been applied to the domain of business work flows. Further investigation is required to determine how portable the approach is to other domains, and if it can deal with specific communicative goals and with different contexts. Taking the *software companies* example from Section 4.3.2, no work has been done to automatically determine when it is most appropriate to use the generic term *companies* or the more specific term *software companies*.

BL aggregation has been studied by Dalianis and Hovy [1996], but it has not been implemented in any actual NLG system, so investigation is still required to determine what data structures are needed to support the approach proposed by Dalianis, and if it will work in practice. Different

approaches could also be explored to logically infer what sets are valid candidates for BL aggregation.

REVISOR's approach to efficient aggregation needs to be attempted on a full-scale aggregation system like STREAK. It has also not been used in a proper opportunistic planning system. It should be carefully compared with CASPER's approach to see which technique provides the best optimizations and to determine if and how they could be combined.

## A A Chronology of Contributions

This section presents a chronological overview of the main contributions that have been made to the topic of aggregation in NLG.

- Hovy [1990b] states the problem of organizing sentence contents. He raises the issues of paraphrasing, determining sentence boundaries and merging propositions through hypotaxis or parataxis.
- Hovy [1990a] includes some work related to UL aggregation in dealing with the rhetorical goals of giving details or interpretations of a situation.
- Scott and de Souza [1990] provide 13 heuristics for applying embedding and paratactic aggregation, as well as for making valid paraphrasing choices. These heuristics are still reflected in recent work.
- Meteer [1991] presents her Text Structure, a representation used to bridge the generation gap between text planning and surface realization, that is the need for text planning to take linguistic constraints into account. She also mentions the Spokesman generation system, which implements Text Structure.
- Horacek [1992] presents an integrated approach to NLG including aggregation.

- Dalianis and Hovy [1993] provide a series of syntactic (paratactic) aggregation rules based on a study on the telephone domain.
- Hovy [1993] provides another good statement of the problem of syntactic (paratactic and hypotactic) aggregation.
- Robin [1994a, 1994b] introduces revision-based opportunistic generation with a working (albeit slow) prototype, STREAK (Surface Text Revisor Expressing Additional Knowledge).
- McKeown *et al.* [1994] and Shaw [1995] present a novel four-step approach to paratactic aggregation, with paraphrasing and elision. This approach is more powerful and more generic than Dalianis's. It is implemented in the prototype application PLANDoc.
- McKeown *et al.* [1995] synthesize her group's work on the use of opportunistic generation in summarization. She presents three linguistic summarization devices, content conflation, syntactic modification and conjunction and discusses conceptual summarization, the task of separating essential and optional information.
- Fiedler and Huang [1995] presents *PROVERB*, which implements the rules of Dalianis and Hovy [1993] in the domain of math proofs, adding some domain-specific aggregation types. *PROVERB* uses Meteer's Text Structure [Meteer, 1991].
- Dalianis [1995a, 1995b] describes implementations of his earlier work in VINST (VIsual and Natural language Specification Tool) and Delphi Tool, a requirements engineering tool.
- Wilkinson [1995] categorizes aggregation on the basis of where it is performed in an NLG system. The paper attempts to clarify issues surrounding aggregation and raises a number of unanswered questions.

- Hovy and Wanner [Hovy and Wanner, 1996, Wanner and Hovy, 1996] repeat the need for microplanning and describe the blackboard architecture used in HealthDoc.
- Fiedler [1996] gives an in-depth study of the microplanner of *PROVERB*, including the aggregation rules it uses, which are very specific to the domain of mathematical proofs, although they are inspired by the work of Dalianis.
- Huang and Fiedler [1996, 1997] describe the use of embedding, grouping, chaining as well as paraphrasing to shorten mathematical proofs in *PROVERB*.
- Passonneau *et al.* [1996] describe FLOWDOC and the ontological subsumption algorithm used to perform unbounded lexical aggregation.
- Robin and McKeown [1996] describe the empirical approach used to build the revision-based summary generation model in STREAK. They describe the corpus used and how it was analyzed, and they show an evaluation of the resulting model.
- Dalianis’s Ph.D. thesis [Dalianis, 1996b] is a collection of papers on aggregation covering paratactic and lexical aggregation and related issues.
- In [Dalianis, 1996a], the introduction to [Dalianis, 1996b], Dalianis defines what aggregation is and classifies the different types of aggregations.
- Dalianis and Hovy [1996] look at lexical aggregation, bounded and unbounded, and how it should be used in conjunction with syntactic aggregation.
- Dalianis [1996c] studies the use of cue words to remove ambiguities introduced by aggregation.

- Reiter and Dale [1997] mention aggregation as one of the standard tasks to include in any NLG system.
- Callaway and Lester [1997] present REVISOR, a scalable revision-based generator. Robin's STREAK is slow because it has to consider too many options and carries too much information around. REVISOR applies its revision operators on an abstract discourse plan containing only the information required to perform the revisions. Using non-monotonic unification, it also has an efficient way of storing and modifying alternative plans, allowing it to quickly search the revision space for the best plan.
- McKeown *et al.* [1997] describe MAGIC, a system using opportunistic generation to generate multimedia health-care briefings. It is reminiscent of PLANDoc.
- Kukich *et al.* [1997] describe the successful reuse of software from PLANDoc to FLOWDoc and finally to ZEDDoc. In its final version, the discourse organizer (the component which performs aggregation and other sentence-planning tasks) is almost entirely domain independent, relying on plug-and-play ontologies to supply the domain-specific information.
- Fehrer and Horacek [1997] use the hearer's domain knowledge to suppress logical steps in mathematical proofs.
- DiMarco *et al.* [1997] show how text selection from a master document yields text which needs repair, including the appropriate application of aggregation.
- Cheng *et al.* [1997] describe how ILEX enhances a draft Text Structure (using Meteer's representation) by use of embedding. She identifies which General Upper Model configurations can be embedded.

Hypotaxis and parataxis are then used to improve the resulting text. This is another example of opportunistic generation.

- Cheng [1998] divides referring expressions in two parts: the referring part identifies the entity, while the non-referring part adds information which is not required for identification purposes, but which the author wants to convey about the entity. The generation of the referring part has been well studied in the past, but the non-referring part has been left out. This paper provides rules to fill this gap.
- Shaw [1998a] describes CASPER's hypotactic aggregation methodology. This implementation is optimized by performing aggregation before lexicalization, with the consequence that the form chosen might not be optimal, but that the system runs much faster. Like Callaway and Lester, Shaw compares his performance with Robin's apparently slow system STREAK.
- Shaw [1998b] expands the work on paratactic aggregation from [Shaw, 1995] in CASPER (**C**lause **A**ggregation in **S**entence **P**lann**ER**), the sentence planner used by PLANDoc and MAGIC.
- Cheng and Mellish [1999] present an experiment run by the authors to determine when embedding is possible for causal and temporal semantic relations, on the basis of human perception of text fluency and content preservation.
- Reape and Mellish [1999] answer many of the questions raised by Wilkinson [1995]. The aggregation classification is further refined within the context of RAGS (A Reference Architecture for Generation Systems), two general senses of aggregation are defined, and the various goals attributed to aggregation by other literature are summarized.

## References

- [Callaway and Lester, 1997] Charles B. Callaway and James C. Lester. Dynamically improving explanations: A revision-based approach to explanation generation. In *IJCAI [1997]*, pages 952–958.
- [Cheng and Mellish, 1999] Hua Cheng and Chris S. Mellish. An empirical analysis of constructing non-restrictive NP components to express two semantic relations. Submitted to *ACL’99*, Maryland, U.S.A. and to *EACL’99*, Norway. Available at [http://www.dai.ed.ac.uk/staff/personal\\_pages/huac/](http://www.dai.ed.ac.uk/staff/personal_pages/huac/), 1999.
- [Cheng *et al.*, 1997] Hua Cheng, Chris S. Mellish, and Michael O’Donnell. Aggregation based on text structure for descriptive text generation. In *Proceedings of the PhD Workshop on Natural Language Generation, 9th European Summer School in Logic, Language and Information (ESSLLI97)*, France, August 1997. Available at [http://www.dai.ed.ac.uk/staff/personal\\_pages/huac/](http://www.dai.ed.ac.uk/staff/personal_pages/huac/).
- [Cheng, 1998] Hua Cheng. Embedding new information into referring expressions. In *Proceedings of the 36th Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (Student Session) (COLING-ACL-98)*, pages 1478–1480, Montreal, Canada, August 10–15 1998. Available at [http://www.dai.ed.ac.uk/staff/personal\\_pages/huac/](http://www.dai.ed.ac.uk/staff/personal_pages/huac/).
- [Dalal *et al.*, 1996a] Mukesh Dalal, Steven K. Feiner, Kathleen R. McKeown, Desmond Jordan, Barry Allen, and Yasser alSafadi. MAGIC: An experimental system for generating multimedia briefings about post-bypass patient status. In *Proceedings American Medical Informatics Association Annual Fall Symposium*, Washington D.C., 1996. Available at <http://www.cs.columbia.edu/~dalal/>.
- [Dalal *et al.*, 1996b] Mukesh Dalal, Steven K. Feiner, Kathleen R. McKeown, Shimei Pan, Michelle Zhou, Tobias Höllerer, James Shaw, Yong Feng, and Jeanne Fromer. Negotiation for automated generation of temporal multimedia presentations. In *Proceedings of the Fourth ACM Multimedia Conference (MULTIMEDIA’96)*, pages 55–64, New York, NY, USA, November 1996. ACM Press. Available at <http://www.cs.columbia.edu/~dalal/>.
- [Dale and Reiter, 1996] Robert Dale and Ehud Reiter. The role of the gricean maxims in the generation of referring expressions. In *Proceedings of the 1996 AAAI Spring Symposium on Computational Models of Conversational Implicature*, Stanford University, California, March 25–27 1996. Available at <http://www.csd.abdn.ac.uk/~ereiter/>.
- [Dale *et al.*, 1990] Robert Dale, Chris S. Mellish, and Michael Zock, editors. *Current Research in Natural Language Generation*. Academic Press, New York, 1990.
- [Dalianis and Hovy, 1993] Hercules Dalianis and Eduard H. Hovy. Aggregation in natural language generation. In Giovanni Adorni and Michael Zock, editors, *Trends in Natural Language Generation: An Artificial Intelligence Perspective*;

*Proceedings of the Fourth European Workshop on Natural Language Generation*, LNAI-1036, pages 88–105, Pisa, Italy, April 28–30 1993. Springer-Verlag. Also published in [Dalianis, 1996b], pages 77–94. Available at <http://www.dsv.su.se/~hercules/>.

- [Dalianis and Hovy, 1996] Hercules Dalianis and Eduard H. Hovy. On lexical aggregation and ordering. In Dalianis [1996b], pages 53–74. Abridged versions published in [EWNLG, 1997], pages 17–27 and in *Demonstrations and posters of the 8th International Workshop on Natural Language Generation, (INLG'96)*, pages 29–32, Herstmonceux, Sussex, UK, June 13–15, 1996.
- [Dalianis, 1995a] Hercules Dalianis. Aggregation, formal specification and natural language generation. In *Proceedings of the First International Workshop on the Applications of Natural Language to Data Bases (NLDB'95)*, pages 135–149, Versailles, France, June 28–29 1995. Also published in [Dalianis, 1996b], pages 105–119. Available at <http://www.dsv.su.se/~hercules/>.
- [Dalianis, 1995b] Hercules Dalianis. Aggregation in the NL-generator of the Visual and Natural language Specification Tool (VINST). In *Proceedings of the Seventh International Conference of the European Chapter of the Association for Computational Linguistics (EACL-95), Student Session*, pages 286–290, Dublin, Ireland, March 27–31 1995. Also published in [Dalianis, 1996b], pages 97–101. Available at <http://www.dsv.su.se/~hercules/>.
- [Dalianis, 1996a] Hercules Dalianis. Aggregation as a subtask of text and sentence planning. In Dalianis [1996b], pages 1–28. Abridged version published in *Proceedings of Florida AI Research Symposium, (FLAIRS-96)*, pages 1–5, Key West, FL, May 20–22, 1996.
- [Dalianis, 1996b] Hercules Dalianis. *Concise Natural Language Generation from Formal Specifications*. Ph.d. thesis, (teknologie doktorsavhandling), Department of Computer and Systems Sciences, Royal Institute of Technology/Stockholm University, June 1996. Teknologie Doktorsavhandling.
- [Dalianis, 1996c] Hercules Dalianis. Natural language aggregation and clarification using cue words. In Dalianis [1996b], pages 123–137. Abridged version published in [EWNLG, 1997], pages 6–16.
- [DiMarco *et al.*, 1997] Chrysanne DiMarco, Graeme Hirst, and Eduard H. Hovy. Generation by selection and repair as a method for adapting text for the individual reader. In *Proceedings of the Workshop on Flexible Hypertext, Eighth ACM International Hypertext Conference*, Southampton, UK, April 1997. Available at <http://logos.uwaterloo.ca/~healthdo/>.
- [EWNLG, 1997] *Proceedings of the Sixth European Workshop on Natural Language Generation*, Duisburg, Germany, March 24–26 1997.
- [Fehrer and Horacek, 1997] Detlef Fehrer and Helmut Horacek. Exploiting the addressee's inferential capabilities in presenting mathematical proofs. In *IJCAI [1997]*, pages 959–964.

- [Fiedler and Huang, 1995] Armin Fiedler and Xiaorong Huang. Aggregation in the generation of argumentative texts. In *Proceedings of the Fifth European Workshop on Natural Language Generation*, pages 5–9, Leiden, The Netherlands, May 20–22 1995. Available at <http://www.ags.uni-sb.de/~afiedler/>.
- [Fiedler, 1996] Armin Fiedler. Mikroplanungstechniken zur Präsentation mathematischer Beweise. Master’s thesis, FB Informatik, Universität des Saarlandes, Saarbrücken, Germany, 1996. Available at <http://www.ags.uni-sb.de/~afiedler/>.
- [Fiedler, 1997] Armin Fiedler. Towards a proof explainer. In *Proceedings of the First International Workshop on Proof Transformation and Presentation*, pages 53–54, Schloss Dagstuhl, Germany, 1997. Available at <http://www.ags.uni-sb.de/~afiedler/>.
- [Fiedler, 1998] Armin Fiedler. Macroplanning with a cognitive architecture for the adaptive explanation of proofs. In INLG [1998], pages 88–97. Available at <http://www.ags.uni-sb.de/~afiedler/>.
- [Hayes and Flower, 1986] John R. Hayes and Linda S. Flower. Writing research and the writer. *American Psychologist*, 41(10):1106–1113, 1986.
- [Horacek, 1992] Helmut Horacek. An integrated view of text planning. In Robert Dale, Eduard H. Hovy, Dietmar Rösner, and Oliviero Stock, editors, *Aspects of Automated Natural Language Generation*, LNAI-587, pages 29–44. Springer-Verlag, Berlin, April 1992.
- [Hovy and Wanner, 1996] Eduard H. Hovy and Leo Wanner. Managing sentence planning requirements. In *Proceedings of ECAI-96 Workshop: New Directions in Planning and Natural Language Generation*, Budapest, August 1996. Available at <http://logos.uwaterloo.ca/~healthdo/>.
- [Hovy, 1990a] Eduard H. Hovy. Pragmatics and natural language generation. *Artificial Intelligence*, 43:153–197, 1990.
- [Hovy, 1990b] Eduard H. Hovy. Unresolved issues in paragraph planning. In Dale et al. [1990], pages 17–45.
- [Hovy, 1993] Eduard H. Hovy. Automated discourse generation using discourse relations. *Artificial Intelligence*, 63:341–385, 1993.
- [Huang and Fiedler, 1996] Xiaorong Huang and Armin Fiedler. Paraphrasing and aggregating argumentative text using text structure. In INLG [1996], pages 21–30. Available at <http://www.ags.uni-sb.de/~afiedler/>.
- [Huang and Fiedler, 1997] Xiaorong Huang and Armin Fiedler. Proof verbalization as an application of NLG. In IJCAI [1997], pages 965–971. Available at <http://www.ags.uni-sb.de/~afiedler/>.
- [IJCAI, 1997] *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, Yokohama, Japan, August 23–29 1997. Morgan Kaufmann.

- [INLG, 1996] *Proceedings of the Eighth International Workshop on Natural Language Generation (INLG-96)*, Herstmonceux, Sussex, UK, June 13–15 1996.
- [INLG, 1998] *Proceedings of the Ninth International Workshop on Natural Language Generation (INLG-98)*, Niagara-on-the-Lake, Ontario, Canada, August 5–7 1998.
- [Kukich *et al.*, 1997] Karen Kukich, Rebecca Passonneau, Kathleen McKeown, Dragomir Radev, Vasileios Hatzivassiloglou, and Hongyan Jing. Software reuse and evolution in text generation applications. In *ACL/EACL Workshop - From Research to Commercial Applications: Making NLP Technology Work in Practice*, Madrid, Spain, July 1997. Available at <http://www.cs.columbia.edu/~radev/>.
- [McKeown *et al.*, 1994] Kathleen R. McKeown, Karen Kukich, and James Shaw. Practical issues in automatic documentation generation. In *Proceedings of the Fourth ACL Conference on Applied Natural Language Processing (ANLP-94)*, Stuttgart, Germany, October 13–15 1994. Association for Computational Linguistics. Available at <http://www.cs.columbia.edu/~shaw/>.
- [McKeown *et al.*, 1995] Kathleen R. McKeown, Jacques Robin, and Karen Kukich. Generating concise natural language summaries. *Information Processing and Management*, 31(5):703–733, 1995. Available at <http://www.di.ufpe.br/~jr/>.
- [McKeown *et al.*, 1997] Kathleen R. McKeown, Shimei Pan, James Shaw, Desmond A. Jordan, and Barry A. Allen. Language generation for multimedia healthcare briefings. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 277–282, 1997. Available at <http://www.cs.columbia.edu/~shaw/>.
- [Meteer and Shaked, 1988] Marie Wenzel Meteer and Varda Shaked. Strategies for effective paraphrasing. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING-88)*, volume 2, pages 431–436, Budapest, Hungary, August 22–27 1988.
- [Meteer, 1991] Marie Wenzel Meteer. Bridging the generation gap between text planning and linguistic realization. *Computational Intelligence*, 7(4):296–304, November 1991.
- [Meteer, 1992] Marie Wenzel Meteer. *Expressibility and the problem of efficient text planning*. Pinter Publishers, London, 1992.
- [Oliver and Hand, 1994] Jonathan J. Oliver and David J. Hand. Introduction to minimum encoding inference. Technical Report TR 4-94, Department of Statistics, Open University, Walton Hall, Milton Keynes, UK, and also TR 94/205, Department of Computer Science, Monash University, Australia, September 1994. Available at <http://www.cs.monash.edu.au/~jono/>.
- [Passonneau *et al.*, 1996] Rebecca Passonneau, Karen Kukich, Jacques Robin, Vasileios Hatzivassiloglou, Larry Lefkowitz, and Hongyan Jing. Generating

- summaries of work flow diagrams. In *Proceedings of the International Conference on Natural Language Processing and Industrial Applications*, University of Moncton, New Brunswick, Canada, June 1996. Available at <http://www.cs.columbia.edu/~hjing/>.
- [Passonneau *et al.*, 1997] Rebecca Passonneau, Karen Kukich, Kathleen R. McKeown, Dragomir R. Radev, and Hongyan Jing. Summarizing web traffic: A portability exercise. Technical Report CUCS-009-97, Computer Science Department, Columbia University, 1997.
- [Reape and Mellish, 1999] Mike Reape and Chris Mellish. Just what is aggregation anyway? In *Proceedings of the Seventh European Workshop on Natural Language Generation*, Toulouse, France, May 13–14 1999. Available at <http://www.itri.brighton.ac.uk/projects/rags/>.
- [Reiter and Dale, 1997] Ehud Reiter and Robert Dale. Building applied natural language generation systems. *Journal of Natural Language Engineering*, 3(1):57–87, 1997.
- [Robin and McKeown, 1996] Jacques Robin and Kathleen R. McKeown. Empirically designing and evaluating a new revision-based model for summary generation. *Artificial Intelligence*, 85(1–2):135–179, 1996. Available at <http://www.di.ufpe.br/~jr/>.
- [Robin, 1994a] Jacques Robin. Automatic generation and revision of natural language summaries providing historical background. In *Proceedings of the 11th Brazilian Symposium on Artificial Intelligence (SBIA-94)*, Fortaleza, CE, Brazil, 1994. Available at <http://www.di.ufpe.br/~jr/>.
- [Robin, 1994b] Jacques Robin. *Revision-based generation of natural language summaries providing historical background: corpus-based analysis, design, implementation and evaluation*. PhD thesis, Columbia University, Computer Science Department, New York, USA, 1994. Available at <http://www.di.ufpe.br/~jr/>.
- [Rosch *et al.*, 1976] Eleanor Rosch, Carolyn B. Mervis, Wayne D. Gray, David M. Johnson, and Penny Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology*, 8:382–439, 1976.
- [Scott and de Souza, 1990] Donia R. Scott and Clarisse Sieckenius de Souza. Getting the message across in RST-based text generation. In Dale *et al.* [1990], pages 47–73. Paper presented at the European Natural Language Generation Workshop, Edinburgh, April 1989.
- [Shaw and McKeown, 1997] James Shaw and Kathleen R. McKeown. An architecture for aggregation in text generation. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), Poster Session*, Yokohama, Japan, August 23–29 1997. Morgan Kaufmann. Available at <http://www.cs.columbia.edu/~shaw/>.

- [Shaw, 1995] James Shaw. Conciseness through aggregation in text generation. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, Student Session*, pages 329–331, 1995. Available at <http://www.cs.columbia.edu/~shaw/>.
- [Shaw, 1998a] James Shaw. Clause aggregation using linguistic knowledge. In INLG [1998], pages 138–147. Available at <http://www.cs.columbia.edu/~shaw/>.
- [Shaw, 1998b] James Shaw. Segregatory coordination and ellipsis in text generation. In *Proceedings of the 36th Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL-98)*, pages 1220–1226, Montreal, Canada, August 10–15 1998. Available at <http://www.cs.columbia.edu/~shaw/>.
- [Wanner and Hovy, 1996] Leo Wanner and Eduard H. Hovy. The healthdoc sentence planner. In INLG [1996]. Available at <http://logos.uwaterloo.ca/~healthdo/>.
- [Wilkinson, 1995] John Wilkinson. Aggregation in natural language generation: Another look. Co-op work term report, Department of Computer Science, University of Waterloo, September 1995. Available at <http://logos.uwaterloo.ca/~healthdo/>.