An experiment in combining two text generators

Laurence Danlos, Guy Lapalme

An experiment in combining two text generators

Laurence Danlos*; Guy Lapalme†
* Université Paris 7
TALANA & LORIA
Campus Scientifique, BP 329,
54506 Vandoeuvre lès Nancy
France
† Département d'informatique et de recherche opérationelle
Université de Montréal
C.P. 6128, Succ Centre-Ville
Montréal, Québec, Canada
H3C 3J7
danlos@linguist.jussieu.fr; lapalme@iro.umontreal.ca

Abstract

In the context of the definition of a reference architecture for NLP, especialy for Natural Language Generation, we describe the problems we encountered when we combined two text generator systems that we initially thought were complementary: one dealing with "what to say?" and the other with the "how to say?".

1 Introduction

In text generation, there is always a problem with the starting point. Although it is relatively easy to evaluate the output by merely reading it and checking if it can be understood, it is much more difficult to characterize the input which can take many forms: objective data, conceptual relations, linguistic informations and, more often, a mix of all these.

One of the best known and often encountered proposition for computing the text structure is the formalism of Rhetorical Structure Theory (RST) (Mann and Thompson, 1988). It was first used and adapted for this purpose by Hovy (1993) and then followed by many other researchers. The rhetorical relations are interpreted as plan operators in a goal-oriented planning paradigm to achieve text structuration. Texts are thus structured in a bottom-up approach contrarily to predefined rhetorical patterns such as the ones used by McKeown (1985).

But some problems have been identified with this view of RST, the most important one being that the original definitions of the relations were open ended leaving much to interpretation; this fact had already been recognized from the start by the authors of RST but to date no other serious "text structure contender" has emerged except for the functional/systemic approach (Matthiessen and Bateman, 1991; Bateman, 1997) which presents completely different method but which has not been demonstrated as appropriate for global text structuring. Other problems with RST relations stem from the fact that they mix semantics and concepts and that they impose a tree structure which is not appropriate for all texts.

Delin et al. (1994) also argue that the RST relations are language specific and that the structure of the text thus practically determines the lexical choices. The rhetorical relations are then merely "nicknames" for linguistic connectors used in the surface structure of the text. (Moore and Pollack, 1992) also argue that RST presumes that there will be a preferred rhetoric relation holding between discourse elements while discourse elements are related simultaneously on multiple levels, e.g. informational and intentional levels.

Some solutions have been proposed to this problem: Kittredge et al. (1991) have proposed adding more precise domain communication knowledge to inform the generator for generating better texts. In the context of instructional texts, Kosseim (1995) has developed the text generator SPIN (Système de Planification d'INtructions) in which an intermediate semantic level is added to avoid the planning of instructions in linguistic terms. Semantic relations are then mapped to rhetorical relations following rules that have been extracted by a corpus study of instructional texts.

In this paper, we describe an experiment that we have done in the context of a collaborative effort between our two laboratories. We took our two existing text generating systems with their respective strengths and we combined them to get a better system.

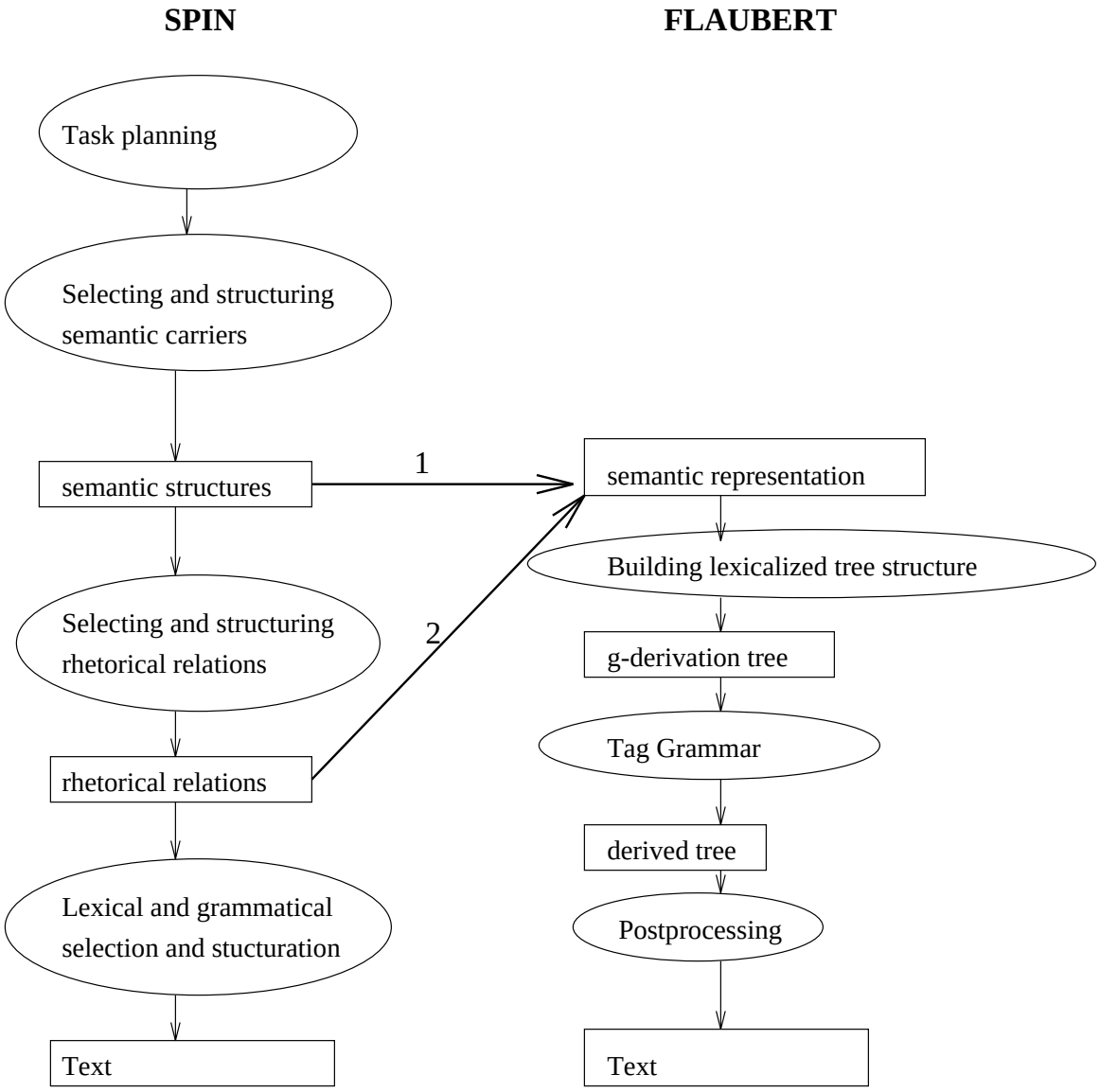

Figure 1: Modules of SPIN and FLAUBERT

2 Description of the two text generators

SPIN (Kosseim, 1995), developed at the Universit´e de Montr´eal, is aimed at the "What to say" part of the generation process namely in the separation of semantic and rhetorical concerns in the context of instructional texts. Danlos and her team (Danlos, 1998; Meunier, 1997), at TALANA in Universit´e Paris 7, developed the FLAUBERT text generator which emphasizes on the "How to say?" part, determining the wording and syntactic constructions using a formalism inspired from TAG called G-TAG (Generation TAG). Although FLAUBERT is domain independent, its first application was for instructional texts. Each system covers the full spectrum of text generation going from a formal input to a well formed natural language output, but each one has its strengths. We decided to combine them to get the "best of both worlds". Other planners or realizers have already been reused before but we are not aware of a previous attempt of combining two **existing** generators into a single one.

So the main problem is then when to stop one system (SPIN) and start the other one (FLAUBERT) in order to get the full benefit of the linguistic basis of Flaubert. Figure 1 shows a schematic of both systems.

Each system has been implemented independently in different programming languages (SPIN in Prolog and FLAUBERT in Ada) and has been developed in a different computing environment but as we will see later, this was not finally the main problem. The details of both systems are not relevant in this discussion, we want to focus on the issues that were raised by this combination. We think that the same questions will arise in any text generators split into modules that we would like to combine either for multi-lingual generation or for sharing work between teams. We had two existing systems and we did not want to simply hack both modules until they had a common interface. We saw this combination as a sort of "psychological study" with two automatic generators that we could study in detail. We think this is the first time such an experiment is reported and is relevant in the context of the definition of a reference architecture for NLP. It also makes a supplementary point for the need for more precise definitions of rhetorical relations.

2.1 Description of SPIN

SPIN (Kosseim, 1995) generates instructional texts from a conceptual representation of the task[1]. It starts from a state such as *The tape cartridge is in the video recorder and it is 10:05* to a goal state such as *A recording for an hour and a half is programmed using an incremental button* and finds a list of instances of primitive operations to go from the start state to the goal state using a non-linear planner. Figure 2 shows what a plan looks like in SPIN. SPIN then chooses semantic carriers using heuristics derived from a corpus study. These heuristics take into account constraints on the cooccurrences of semantics carriers, knowledge and intention of the reader as perceived by the writer and the communication goal. After this step, we get a list of semantic carriers as shown in Figure 3.

These semantic carriers are then translated into rhetorical relations, shown in Figure 4, using other corpus deduced heuristics that depend on the cooccurrence constraints of rhetorical relations, on the knowledge and intention of the reader and the nature of the task. The rhetorical structure is finally transformed into a well formed French text.

2.2 FLAUBERT

FLAUBERT takes as input a graph of events built by the user who fills a questionnaire through an interface that proposes cascading menus based on a domain model. The system does not have a planning stage nor a user model, the emphasis is put on linguistic questions such as lexical choices including choices of connectors, length and content of sentences and clauses, parallellism and stylistic issues. FLAUBERT uses three databases:

- a domain model describing an ontology of concepts in a typed feature formalism, the concepts include objects, actions, states and relations between them;
- a set of lexical data bases associated with concepts; for a given concept, its lexical database describes its semantico-lexical realizations (lexical heads and argument structures) accompanied with tests of applicability for right semantics and well formedness;
- a TAG grammar whose syntactic informations allow a derived tree to be computed from a derivation tree.

FLAUBERT can generate well formed English and French texts. An excerpt of the event graph corresponding to the VCR recorder example used in the preceding section is given in Figure 5.

The g-derivation tree built from this event graph is given in figure 6 and the English generated text is:

```
To get a better image quality, set the
speed selector to SP. Next select
the channel 4. Then ... . Finally, ....
```

3 Combining SPIN and FLAUBERT

An "obvious" interface between SPIN and FLAUBERT is at the semantic level where a semantic structure of SPIN and an event graph of FLAUBERT must be matched (see arrow labeled 1 in figure 1) [2] but this posed problems such as:

[1] SPIN generates French text but, in this paper to simplify the presentation, we have translated the examples and the formalism in English while keeping the original idea

[2] We also experimented by connecting our two systems by mapping the rhetorical relations of SPIN (arrow 2 of Figure 1) to the semantic

```
set speed selector(precond: better quality, success: speed selector set}
touch channel selector (success: channel changed,
                        fail   : not (channel changed))
push OTR button 1 times (success: PM 10:35 (30 min),fail: PM 10:05)
push OTR button 2 times (success: PM 11:05 (1 h),fail: PM 10:35 (30 min))
push OTR button 3 times (success: PM 11:35 (1h30 min),fail: PM 11:05 (1 h))
push TIMER button within 9 sec (success: recording on, fail: not (recording on))
```

Figure 2: Plan for programming a VCR in SPIN

```
title(program(obj:recording, manner: incremental button))
option(better image quality,set(obj:speed selector, dest: SP))
op_seq(set(obj:speed selector, dest: SP))
op_seq(set(obj:channel 4))
op_seq(push(dest: OTR button))
op_seq(push(dest: TIMER button, manner: within 9 sec))
```

Figure 3: Semantic carriers for the VCR example chosen by SPIN

```
title
goal(program(obj:recording, manner: incremental button),[])
paragraph
goal(better image quality), comma
action([],set(obj:speed selector, dest: SP)), period
action([],set(obj:channel 4)), period
action([],push(dest: OTR button)), period
action([],push(dest: TIMER button, manner: within 9 sec))
```

Figure 4: Rhetorical relations for the VCR example determined by SPIN

- SPIN uses a conceptual representation of the task: simple objects are represented as character strings and operations are predicates on these objects; in FLAUBERT, the objects are entities whose representation must be deduced from a typed attribute structure
- as the emphasis in SPIN had been put on "what to say?", some semantic carriers are given in an almost surface form like

  `on_an_incremental_button`

  FLAUBERT is more linguistically oriented and rigorously separates predicates, arguments and modifiers.

elements of FLAUBERT but then FLAUBERT was not very useful because all the linguistic choices for which it has been designed to do had already been done by SPIN.

Given suitable "unix scripts", we could (and we did in fact) manage to connect these two systems to produce some text but this approach was not satisfactory because more fundamental problems were raised.

Taking as a stopping point the semantic structure of SPIN means that an equivalent of the "choice of a rhetorical structure" of SPIN be done in FLAUBERT when expressing a semantic relation. For example, in SPIN, an OUTCOME relation can be translated into a rhetoric relation of *goal*, *manner* or *result*, each of these could be associated with one or two connectors while in FLAUBERT, the OUTCOME relation is directly associated with a set of connectors. In SPIN, this choice depends, among other things, on the fact that it is an intended outcome or not.

For SPIN, the two following texts can be generated from the single OUTCOME relation

1. *To shut the radio off, turn the volume button com-*

```
E0  := Instruction [ title    => E1
                     sequence => E2 ]
E1  := Title [ Action => E11 ]
E11 := Program [ object => recording
                 with   => an_incremental_touch ]
E2  := Sequence [ first => E21
                  second=> E22]
E21 := Option [ goal   => better_image_quality
                manner => E211 ]
E22 := Sequence [ first => E221
                  second=> E222 ]
E221:= Select [ object => channel_4 ]
         ...
```

Figure 5: Rhetorical relations for the VCR example determined by FLAUBERT

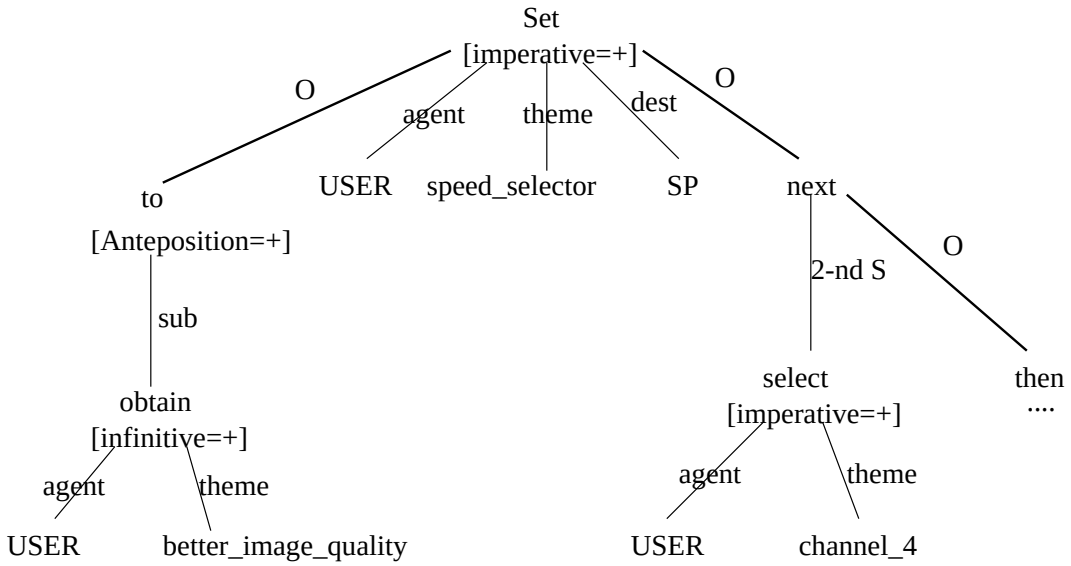

Figure 6: G-derivation tree built from the event graph of Figure 5

pletely to the left.

2. *When you stop the motor of the car, the radio will be turned off*

In (1), a desirable outcome is associated with *imperative + to infinitive*, while in (2), an undesirable outcome, is associated with a juxtaposition and a comma. This choice depends on semantic criteria for which FLAUBERT is not designed for. So although an OUTCOME relation can be defined and used, additional information has to be given in the form of additional features in the event description, like

```
OUTCOME < binary_event
       [ action ==> simple_event
         outcome==> simple_event
         desirable ==> yes/no    ]
```

But this approach is not really appropriate for FLAUBERT. It would be better to separate OUTCOME into two different relations, e.g. INSTRUCTION for (1) associated with the *imperative + to infinitive* and with a few other structures of connector, and SIDE-EFFECT for (2) associated with juxtaposition.

This difference between SPIN and FLAUBERT reflects a fundamental question both in natural language processing and in knowledge representation: do those two texts really represent the same conceptual relations?

	SPIN	Hovy (1993)	FLAUBERT	Delin et al. (1994)
Planning	+	+	-	+
Conceptual structure				
Gross level	+	+	-	-
Fine level	-	-	+	+
Rhetorical structure of target language	+	+	-	?
Text	+	+	+	?

Table 1: Comparison of 4 generators; + or - indicates that the representation is used or not; ? indicates that we are not sure if it is used.

Thus we can see that a semantic relation in a system has not exactly the same meaning in another, this is a good illustration of the *generation gap* best described by Meeter (1991) where a text planner has to be aware of the linguistic consequences of the choices it made. So it is difficult to separate the generation process into independent modules; but nevertheless nearly all generation systems do make that division (Reiter, 1994). Table 1 compares 4 different levels of representation computed in the process of text generation. It shows that the systems that use a conceptual structure at a gross level (SPIN and Hovy) use a rhetorical structure, whereas the systems that use a conceptual structure at a fine level (FLAUBERT and Delin) do not need a rhetorical structure. A gross level conceptual relation like OUTCOME covers various situations, e.g. (1), and (2) and why not (3) which is quite plausible in a war context.

3. *A bomb blew up my car, turning off the radio.*

Therefore an intermediate level (e.g. a rhetorical structure) is needed before choosing appropriate connectors. On the other hand, a fine level conceptual structure like INSTRUCTION in FLAUBERT or GENERATION in Delin is associated with a set of two or three connectors which require only to be accompanied with syntactic and lexical tests. Let us add that fine level conceptual relations can be even more precise that rhetorical relations. Consider sentences (1) given above and (4):

4. *Ted folded up the sheets to please Mary.*

In RST, it would be said that both (1) and (4) involve a PURPOSE relation. This amounts to give a nickname to the structure of connectors formed with *to infinitive*. With fine level conceptual relations, (1) and (4) come under two different relations because they come under two different textual genres, instructional text for (1) and narrative text for (4). This discrepancy between textual genres implies a basic difference in the semantics of *to infinitive*: in (1), it is true that the radio is shut off every time the user turns the volume button completely to the left except when it is broken; let us say that the goal is systematically achieved. While in (4), nobody knows if the goal is achieved: Mary might not be pleased at all by Ted's folding up the sheets because, for example, she wanted to iron them before. This basic semantic difference can be reflected with the use of two fine level conceptual relations: INSTRUCTION or GENERATION, a procedural relation in instructional texts, for (1), and GOAL in narrative texts for (4). In summary, if fine level conceptual relations are more precise than RST relations, there is clearly less need for a rhetorical level. But the question remain as if it is feasible to do task planning in terms of fine level relations.

This experiment shows that rhetorical relations are often defined differently or at different levels in various generating systems. We have documented here this case study of a concrete attempt at combining two generating systems but we are convinced that we would have faced the same problems by trying to link other generators. This could be interpreted as a failure because it made shortcomings of SPIN stand out but we thought it might be interesting for others to be aware of this aspect that seems to have been hidden because either the deep generator or the surface generator module was being forced into giving of accepting the appropriate level of input. This should prompt the generation community to find a more precise definition of rhetorical relations that the ones currently used. This step is an unavoidable one if we want to be able to re-use or share the work done in other teams.

4 Conclusion

This experiment thus shows the need for a clarification of the so called "rhetorical" level: is it more than a taxonomy of connectors? how does it interfere with lexical choices? is it language independent? what should be the level of definition of a rhetorical relation? Should we want the text generators to be able to build on the works of others we will need methods for comparing, evaluating and combining generators based on a more precise definition of a semantic or rhetorical relation. This experiment did not give a definite answer to these questions but it illustrates the somewhat arbitrary character of semantic relations currently used in natural language generation. So any reference architecture definition should make sure that these concerns are addressed in an appropriate manner.

Acknowledgment

We thank the "Programme Franco-Qu´eb´ecois de coop´eration scientifique et technique, thème Ing´enierie linguistique de la connaissance" for granting this research. We also thank Leila Kosseim for her collaboration and the students of TALANA who carried out the experiments: Fr´ed´eric Meunier, Marie-Pierre Delaunay and Nadjet Bouayad-Agha.

References

J. Bateman. Enabling technology for multilingual natural language generation: the KPML development environment. *Natural Language Engineering*, 3(1):15–55, 1997.

L. Danlos. G-TAG : un formalisme lexicalis´e pour la g´en´eration de textes inspir´e de TAG. *T.A.L.*, 39(2):27 p., 1998.

J. Delin, A. Hartley, C. Paris, D. Scott, and K. Vander Linden. Expressing Procedural Relationships in Multilingual Instructions. In *Proceedings of the* 7^{th} *International Workshop on Natural Language Generation*, pages 61–70, Kennebunkport, Maine, 1994.

E. H. Hovy. Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63(1–2):341–385, 1993.

R. Kittredge, T Korelsky, and O. Rambow. On the need for domain communication knowledge. *Computational Intelligence*, 7(4):305–314, 1991.

L. Kosseim. *Planification de textes d'instruction: Sélection du contenu et de la structure rhétorique*. PhD thesis, Universit´e de Montr´eal, 1995.

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

C. Matthiessen and J. Bateman. *Text Generation and Systemic-Functional Linguistics*. Pinter, London, 1991.

K. R. McKeown. *Text generation, Using discourse strategies and focus constraints to generate natural language text*. Studies in Natural Language Processing. Cambridge University Press, 1985.

M. Meeter. The generation gap. *Computational Intelligence*, 7(4):296–304, 1991.

Fr´ed´eric Meunier. *Implantation du formalisme de génération G-TAG*. PhD thesis, Universit´e Paris 7, 1997.

J. Moore and M. Pollack. A Problem for RST: The Need for Multi-Level Discourse Analysis. *Computational Linguistics*, 18(4):537–544, 1992.

Ehud Reiter. Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible? In *Proceedings of the Seventh International Workshop on Natural Language Generation*, pages 163–170, Nonantum Inn, Kennebunkport, Maine, 1994.