Aggregation and Contextual Reference in Automatically Generated Instructions

Geert-Jan M. Kruijff & Ivana Kruijff-Korbayová*

Institute of Formal and Applied Linguistics (ÚFAL)
Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic
{gj, korbay}@ufal.mff.cuni.cz

1 Planning textual coherence and cohesion

The quality of *any* text is for an important part determined by its *coherence* and *cohesion*. Informally, a coherent text makes sense and is intelligible for a reader, while a cohesive text is economic and sticks together. Fig. 1 shows three short instruction texts which convey the same content, but differ in coherence and cohesion: (1) illustrates a coherent but minimally cohesive version, characteristic by many repetitions; (2) illustrates cohesive but rather incoherent versions, where incoherence is due to inappropriate pronominalization; finally, (3) illustrates coherent as well as cohesive versions.

Both coherence and cohesion are a matter of degree. A good quality text can be characterized by exhibiting maximal coherence and maximal cohesion at the same time. We consider (3i) less cohesive than (3ii), because the latter version has a smoother thematic progression, and consequently allows pronominalization in the last sentence. Precise and general quantification of coherence and cohesion is, however, rather difficult.

The principal means increasing text cohesion are *aggregation* and *contextual reference*. Aggregation involves expressing multiple pieces of content in a compact way using linguistic expressions of higher complexity, e.g. one complex sentence rather several simplex ones, or complex nominal groups involving disjunction or conjunction. Contextual reference means using contextually suitable forms of abbreviated reference to entities which either have already been mentioned or are related to already mentioned entities by some inferrable relation.

We employ the working hypothesis that the *goal* of aggregation and contextual reference is *increasing text cohesion*, while *not decreasing text coherence*. We discuss how to make this goal more precise, in order to be able to employ it in a multilingual system for automatic generation of instructions [7, 6].

The planning of aggregation and contextual reference starts from considering how the content is structured and how its pieces are related. This leads to the question where the planning of aggregation and contextual reference should be

* The work reported here was carried out in the context of the AGILE project, supported by the European Commission within the COPERNICUS programme, grant No. PL961104. We would like to thank John Bateman for extensive discussion.

(1) *Coherent but minimally cohesive*
Draw a line segment:
First specify the start point of the line segment.
Then specify the end point of the line segment.
Draw an arc segment:
First switch to the ARC mode. Enter a.
Then snap the arc to the end point of the line segment:
Enter endp. Select the line segment.
Finally, specify the end point of the arc segment.

(2) *Cohesive but rather incoherent*
To draw a line segment specify the start point and end point of the line segment.
Draw an arc segment:
First switch to the ARC mode by entering a.
i. Then in order to snap it to the end point of the line segment,
enter endp and select it.
ii. Then snap the arc segment to the end point of the line segment
by entering endp and selecting it.
Finally specify { the end point of the arc segment | its endpoint }.

(3) *Coherent and cohesive*
Draw a line segment by specifying its start point and end point.
Draw an arc segment.
First switch to the ARC mode by entering a.
i. Then enter endp and select the line segment
in order to snap the arc segment to the end point of the line segment.
ii. Then snap the arc segment to the end point of the line one
by entering endp and select the line segment.
Finally specify the end point of the arc segment.

Fig. 1. Differences in coherence and cohesion in instruction texts

placed relative to the reference architecture for NLG systems [8]. This architecture consists of a set of modules, set in a sequence. First comes the specification of the content that is to be realized by the text. Given that content, a *text planner* creates a *text plan* (TP) that organizes the content into such a structure that it becomes possible to realize the content in a linear fashion. Subsequently, a *sentence planner* interprets the TP, and generates sentence plans (SPs) that specify how to realize bits of content identified by the TP by (possibly complex) sentences. Finally, these SPs are fed sequentially into a *lexico-grammar* that generates the individual sentences, the sequence of which makes up the text realizing the content that was served at the beginning.

To deal with the issues of aggregation and contextual reference we propose to work with a two-stage process. First, aggregation and contextual reference *as a means to increase cohesion* are planned within **text planning**, because of their effect on establishing a thematic progression, which concerns larger text units than a single sentence. Thematic progression consequently influences ordering -

primarily, "new" information is to be ordered towards the end of the sentence.[1] Second, aggregation and contextual reference *as a linguistic form of expression* are planned at the level of **sentence planning**. There, on the basis of the TP we specify or constrain the form used to realize the content. These constraints are still, as far as possible, relatively language-independent, except where it concerns language-specific realization of particular concepts (cf. the discussion in [6]). As a consequence, particular aspects of cohesion are left to the individual lexico-grammars. For example, some decisions that lead to increased cohesion are a matter of linguistic realization. Notably, this concerns the "deletion"[2] of material which would otherwise result in repetition. The conditions for deletion are language-specific, and deletion can be restricted by particular syntactic contexts. Therefore, we conceive of deletion as a lexico-grammatical realization issue, and not one of sentence planning.

Abstractly put, a contextually appropriate content realization should take the form of a *minimally complex, and maximally informative (or maximally individuating) expression*. For aggregation, this means realization by a suitable complex expression. With regard to clause complexes, a decision needs to be made, for example, between different types of semantic relations (e.g., Purpose vs. Means), between ways of realizing these and of realizing the relata themselves (e.g. as hypotactic clause complex or as a clause and a nominalization).

For contextual reference, maximal informativity means that, ideally, an expression should refer unambiguously. At the same time, minimal complexity requires a form as simple as possible. The choice of the exact linguistic form depends on the presence of "competitors" in the context. For example, in the context established in (3), there is a mode, a line segment and an arc segment. In order to determine a particular segment in a possible sequel, one would utter either 'the line segment' or 'the arc segment'. In another sequel where attention was already centered on segments, one could refer by the more cohesive expressions 'the line one' or 'the arc one'. Moreover, to refer to a highly activated entity, for instance the most recently mentioned segment, one can use a pronoun.

Below we elaborate on these ideas, discussing how they have been implemented in a multilingual system for automated text generation [7, 6]. We focus on the module responsible for text- and sentence-planning, an earlier version of which we discussed in for example [3].

2 Text and sentence planning in AGILE

Our text- and sentence planning module (TSM) is capable of planning a plethora of text types as found in software manuals, and one or more styles within each

[1] The tag "new" is not entirely accurate, even though it is often used: Information can be either new in the sense of *adding* information, or it can be new in the sense of *modifying* information that had already been available in the context. Cf. the discussion of handling word order in AGILE in [5].

[2] Cf. the literature on phenomena like *deletion, ellipsis, gapping, stripping, etc.*.

text type. In this paper, we only consider the text type of *full procedural instructions*, on which we illustrate the TSM's capabilities in planning different types of aggregation and contextual reference to enhance coherence and cohesion.[3]

In order for a reader to easily understand how to accomplish the task to which the instructions pertain, the text needs to reflect in a perspicuous fashion the hierarchical organization of the task at hand as well as the sequence of steps to be carried out. The hierarchical organization of the tasks is reflected straightforwardly by the hierarchical organization of the text, or by the hypotactic and paratactic relationships between clauses within clause complexes. The former is an issue of layout, while the latter is a part and parcel of aggregation. Below, we broadly consider three different classes of aggregation: discourse aggregation (rhetorical relations like Purpose or Means), syntactic aggregation (conjunction or disjunction), and explicit sequence marking.

The main characteristic of the cooperation between the phases of text planning and sentence planning in the AGILE TSM is that the TP contains *all* the decisions how to relate the realizations of various bits of content. The sentence planner only *interprets* these decisions, as reflected in the TP, and creates SPs that state how to realize these relations.

Discourse aggregation In the task-oriented setting of AGILE, discourse aggregation is formulated as handling the relation between two tasks where one of them corresponds to a goal, and the other is a step within a method accomplishing or contributing to accomplishing the goal. Prototypically, such content configuration yields the RST relations of Purpose and Manner.

We regard the distinction between these two relations in terms of the difference between "enablement" and "generation" between processes ([**?**])). Generation holds between two processes a and b if the performance of a automatically brings about the occurrence of b. Enablement holds between a and b if the execution of a brings about a set of conditions that are necessary, but not necessarily sufficient for the subsequent performance of b. In our TPs, enablement is modeled by Purpose, and generation by Manner. The distinction is made on the basis of the underlying concepts. As a simplification, we treat simple physical actions as generating and higher level processes as enabling.

Syntactic aggregation Syntactic aggregation also takes place between tasks, but in this case not between hierarchically related tasks, but rather between sequentially related ones or between alternatives accomplishing the same goal. As steps in a PROCEDURE-LIST constitute a sequence, they can be related through conjunction with or without additional explicit sequence marking, whereas methods in a METHOD-LIST are alternatives, and therefore can be listed in a disjunction. In both cases, impose complexity constraints on the tasks that can be related through either conjunction or disjunction.

[3] The other text types the TSM is capable of planning are summaries/"cheat sheets", functional descriptions, tables of content, and overviews.

The TP in Figure 2 contains an RST-purpose relation, and several conjunctions. The CONJOINED-INSTRUCTION-TASKS nodes indicate that the left-daughter node (a task) and the task dominated by the immediate non-terminal node above a CONJOINED-INSTRUCTION-TASKS node, are to be related by a conjunction. Hence, the TP specifies *three* tasks to be aggregated in that way.

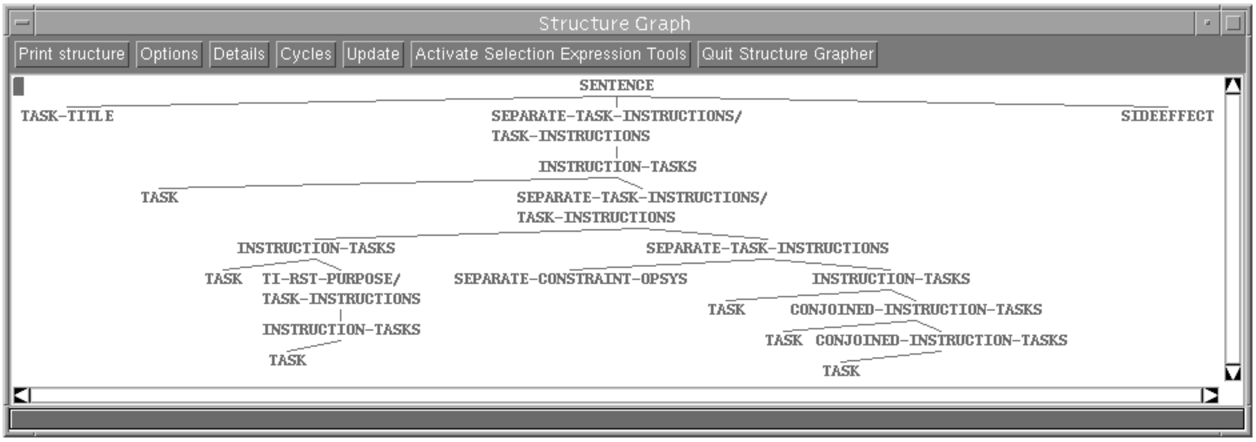

Fig. 2. Text plan showing conjunction and RST-purpose

Besides relations like RST-purpose and RST-manner, or conjunction and disjunction, we also plan sequence realization. The straightforward way of reflecting the order of steps is by the corresponding ordering of sentences in the output. However, when the content is complex, more elaborate structuring of the generated text becomes needed as a means to facilitate a reader's orientation. We employ the following repertoire of sequence realization styles:

Running text sequence : *unmarked* (continuous paragraph) or *linguistically marked* (continuous paragraph including overt sequence discourse markers)
List sequences : *unmarked* (bullet list), *numbered* (numbered list) or *linguistically marked* (bullet list including overt sequence discourse markers)

Sequence discourse markers are expressions like 'first', 'further', 'finally', 'now', 'then', etc. (and their Bulgarian, Czech and Russian counterparts).

Sentence Planning Each of the leaves in the TP (like the tasks, the Task-Title, and the Side-effect in Figure 2) is linked to a bit of content in the content specification. For more discussion, see [3, 6].

The sentence planner then performs its job as follows. It begins by traversing the TP, and creating SPs specifying how the content of each leaf should be realized as a single clause. While recursively traversing the TP depth-first manner, the sentence planner leaves *traces* that indicate whether the sentence planner has encountered any nonterminal node specifying a particular type of relation (e.g., conjunction). Once the sentence planner has created SPs for all the leaves, the recursion is reversed, and all the SPs are collected into a large sequence of SPs. It is at this point of "collecting the SPs" that the traces are being processed. When the sentence planner encounters a trace, then we add to the sequence of plans a *complex* sentence plan specifying the two clauses *and* their relation *as a clause complex*, and not the sentence plans for the individual clauses themselves.

Thus, the sentence planner only *interprets* the TP —it does not make any independent decisions about how to relate clauses. As we have already seen, this places all the responsibility for planning such relations with the text planner. This is an important point, for this means that we can effectively write *text grammars* just like we write lexico-grammars; only, in the case of text grammars we specify how to generate texts, not individual sentences. Together with the fact that the existing sentence planner can easily be extended to cover any type of relation distinguished by the text planner, we obtain a perspicuous and principled way in which we can plan and realize textual coherence and cohesion.

A final word should be said about contextual reference and information structure. The space-limits of the current paper do not permit us to elaborate much on the approach we take (but see [5]). In a nutshell, when processing the TP we build a discourse model that keeps track of what bits of content have already been used (i.e. introduced in the discourse). Whenever such content is re-used, the sentence planner specifies it as *contextually bound* [9]. Subsequently, the uses this information to generate contextually appropriate word order, as discussed in [5] and inspired by [9]. In the future, we hope to extend this approach adopting the proposals discussed in [1].

References

1. J. A. Bateman. 1999. Using aggregation for selecting content when generating referring expressions. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 127–134, Maryland, United States of America.
2. J. Delin, T. Hartley and D. Scott. 1996. Towards a contrastive pragmatics: syntactic choice in English and French instructions. in *Language Sciences* 18 (3-4): 897-931.
3. G.J.M. Kruijff and I. Kruijff-Korbayová. 1999. Text structuring in a multilingual system for generation of instructions. In [4].
4. *Proceedings of Text, Speech and Dialogue '99*, Czech Republic. Springer Verlag.
5. I. Kruijff-Korbayová and G.J.M. Kruijff. 1999. Handling Word Order in a Multilingual System for Generation of Instructions. In [4].
6. G.J.M. Kruijff *et al.* 2000 Multilinguality in a text generation system for three Slavic languages In *Proceedings COLING 2000.*
7. I. Kruijff-Korbayová. 1998. Generation of instructions in a multilingual environment. In Petr Sojka, Václav Matoušek, Karel Pala, and Ivan Kopeček, editors, *Text, Speech and Dialogue (Proceedings TSD'98)*, Brno, Czech Republic, September.
8. E. Reiter and R. Dale. 2000. *Building Natural-Language Generation Systems.* Cambridge University Press, Cambridge, England.
9. P. Sgall, E. Hajičová and J. Panevová. 1986. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects.* D. Reidel Publishing Company, Dordrecht, Boston, London.