Contents

Analyzing the explanation structure of procedural texts: dealing with Advices and Warnings

Lionel Fontan
Patrick Saint-Dizier
IRIT, Toulouse
email: stdizier@itit.fr

Abstract

We present the explanation structure of procedural texts, focusing on the analysis of argumentation, in particular advices and warnings. Their role is to support and motivate the goal-instruction structure, which is the skeleton of procedural texts. Procedural texts consist of a sequence of instructions, designed with some accuracy in order to reach a goal (e.g. assemble a computer). Procedural texts may also include subgoals. Goals and subgoals are most of the time realized by means of titles and subtitles. The user must carefully follow step by step the given instructions in order to reach the goal. We introduce the notion of *instructional compound*, which is a complex structure that articulates instructions with various discourse elements. We then show how arguments can be extracted with the help of patterns. We finally investigate their contribution to the structure and the understanding of procedural texts.

1 Introduction

The main goal of our project is to analyze the structure of procedural texts in order to efficiently and accurately respond to How-to-do-X? questions. This means identifying titles (which convey the main goals of the procedure), sequences of instructions serving these goals, and a number of additional structures such as prerequisites, warnings, advices, illustrations, etc. (Takechi et al. 2003, Adam, 2001). A response to an How-to question is then the well-formed text portion within the scope of the title that matches the question.

In our perspective, procedural texts range from apparently simple cooking recipes to large maintenance manuals. They also include documents as diverse as teaching texts, medical notices, social behavior recommendations, directions for use, assembly notices, do-it-yourself notices, itinerary guides, advice texts, savoir-faire guides etc. (Aouladomar et al., 2005). Procedural texts follow a number of structural criteria, whose realization may depend on the author's writing abilities, on the target user, and on traditions associated with a given domain. Procedural texts can be regulatory, procedural, programmatory, prescriptive or injunctive. The work we report here was carried out on a development corpus of 1,700 French texts taken from the Web from most of the areas cited above, and extracted randomly from our more global corpus of 8,000 texts.

We have developed a detailed analysis of procedural texts from a manual corpus analysis, identifying their main basic components as well as their global structure. For that purpose, we have defined two levels: a segmentation level that basically identifies structures considered as terminal structures (titles, instructions, prerequisites, connectors, etc.) and a grammar level that binds these terminal structures to give a global structure to procedural texts (Delpech et al. 2008). This structure is textual and dedicated only to elements relevant to procedurality. To design the grammar, we have borrowed a few considerations from Minimalist syntax, in particular recent developments of X-bar syntax and notions like merge. This point will not be developed here (but see Delpech et al. 2007). Note that text grammars are generally in a relatively early development stage (Webber 2004, Gardent 1997).

Procedural texts are complex structures, they often exhibit a quite complex rational (the instructions) and 'irrational' structure which is mainly composed of advices, conditions, preferences, evaluations, user stimulations, etc. They form what we call the explanation structure, which motivates and justifies the goal-instructions structure, viewed as the backbone of procedural texts. A number of these elements are forms of argumentation, they appear to be very useful, sometimes as important as instructions, they provide a strong and essential internal cohesion and coherence to procedural texts.

An important aspect of this project is the accurate identification of the

explanation structure as found in procedural texts in order to better understand explanation strategies deployed by humans in precise, concrete and operational situations and to transpose it to the generation of explanations in general and in cooperative question-answering systems in particular. We have already studied the instructional aspects of procedural texts and implemented a quite efficient prototype within the TextCoop project (Delpech et al. 2008) that tags titles and instructions with dedicated XML tags. In this paper, after a categorization of explanation structure as found in our corpus of procedural texts, and the presentation of instructional compounds, we focus on the recognition of warnings and advices. The work is realized for French, examples in English are just glosses.

2 The explanation structure in procedural texts

We will introduce two independent structures: the general organization of the explanation structure, and the notion of instructional compound.

2.1 A global view of the explanation structure

Procedural texts have a very rich semantic structure, with a large number of facets. From our development corpus (1,700 web texts of 1–3 pages), we established a classification of the different forms explanations may take. The explanation structure is meant to guide the user in two ways: (1) by making sure that he will effectively realize actions as they are specified, via arguments (Amgoud et al. 2001, 2005) such as threats, rewards, advices and warnings which are 'coercitive' in a certain sense, and (2) help considerations such as evaluation of work realized so far and encouragements of different kinds.

The main structures we have identified in this type of text are facilitation and argumentation structures. They are either global (adjoined to goals, and having scope over the whole procedure) or local, included into instructional compounds, with a scope local to the instructional compound (see next section). The latter is by far the most frequently encountered case.

Explanation structures can be organized as follows (the terminology is borrowed from existing work on rhetorical relations or introduced by ourselves):

- **facilitation structures**, which are rhetorical in essence (Kosseim et al. 2000, Vander Linden 1993), correspond to *How to do X?* questions, these include two subcategories:
  1. user help, with: hints, evaluations and encouragements;
  2. controls on instruction realization, with two cases:
     (a) controls on actions: guidance, focusing, expected result and elaboration;

(b) controls on user interpretations: definitions, reformulations, illustrations and also elaborations.

- **argumentation structures**, corresponding to *Why do X?* questions. These have either:

  1. positive orientation with the author involvement (promises) or not (advices and justifications), or
  2. negative orientation with the author involvement (threats) or not (warnings).

In what follows, we will mainly concentrate on this second point, and in particular on warnings and advices which are the most frequently encountered arguments (since there are rarely involvements from the author). Roughly, we have about 25% of instructions which have arguments in do-it-yourself texts, and up to 60% in social procedural texts. Argumentation structures are relatively general to an application domain, while facilitation structures are much more specific to the text and the targeted audiences.

2.2 From instructions to instructional compounds

In most types of texts, we do not just find sequences of simple instructions but more complex compounds composed of clusters of instructions, that exhibit a number of semantic contextual dependencies between each other, that we call **instructional compounds**. These compounds are organized around a few main instructions, to which a number of subordinate instructions, warnings, arguments, and explanations of various sorts may possibly be adjoined. All these elements are, in fact, essential in a compound for a good understanding of the procedure at stake.

An instructional compound has a very rich, but relatively well organized, discourse structure, composed of several layers, which correspond to the different aspects instructions may have. The structure is the following:

- The **goal and justification** level, which has in general wider scope over the remainder of the compound. It indicates motivations for doing actions that follow in the compound (e.g. *You must regularly clean the curtains of your bedroom: to do this ...; To change your mother card, you must...*, which here motivates actions to undertake). It gives the fundamental motivation of the compound. Compared to titles, these introduce very local goals. These are not considered in the goal hierarchy introduced by titles, and they will not be considered for question answering.

- The **instruction core (or kernel) structure**, which contains the main instructions. These are more or less explicitly temporally organized (see

below). In general simple sequentiality prevails, the goal being to limit the intellectual load imposed to the user. Actions are identified most frequently via the presence of action verbs (in relation to the domain) in the imperative form, or in the infinitive form introduced by a modal (Delpech et al. 2008). We observed also a number of subordinated instructions forms adjoined to the main instructions. These are in general organized within the compound by means of rhetorical relations, as the examples below will illustrate.

- The **deontic and illocutionary force structures**: consist of marks that operate over instructions, outlining different parameters. These linguistic structures play a major role in argumentation:
  - deontic: obligatory, optional, forbidden or impossible, alternates (or),
  - illocutionary and related aspects: stresses the importance of actions: necessary, advised, recommended, to be avoided, etc. These marks are crucial to identify the weight of an argument.
- a **temporal structure** that organizes sequences of instructions (and, at a higher level, instructional compounds). In general, the temporal structure is simple, sequentiality prevails. In some cases, parallel actions are specified, which partially overlap. Action verbs may indicate some form of parallelism of actions (*incorporate and stir*). In other cases it is the user's experience that determines the exact temporal interpretation.
- The **conditional structure**: introduces conditions over instructions within the compound or even over the whole instructional compound. We encounter quite a lot of structures organizing mutually exclusive cases *If you are pregant, take medicine X ..., if you are not pregnant and between 19 and 65, it is advised that you take medecine Y, ...*
- The **rhetorical structure** whose goal is to enrich the core instructions by means of a number of subordinated aspects (realized as propositions, possibly instructions) among which, most notably: enablement, basic forms of motivation, circumstance, elaboration, instrument, precaution, manner. A group of relations of particular interest in this paper are arguments, developed hereafter.
- The **causal structure** indicating the goal of an action. We identified in procedural texts four types of causal relations, following (Talmy 2001): intend-to (direct objective of an action: *push the button to start the engine*), Instrumented (*use a 2 inch key to dismount the door*), Facilitation (*enlarge the hole to better empty the tank*) and Continue (*keep the liquid*

warm till its color changes). These are local to a single instruction, with no a priori interaction with the goal or justification level advocated above.

Explanations and arguments help the reader understand why an instruction must be followed and what are the risks or the drawbacks if he does not do it properly. We will illustrate this with two instructional compound examples. In the first example, we have three main instructions, and an elaboration is adjoined to the second one. The temporal relations are simple (realized by *then*, *and*) and are not represented:

$[_{instructional-compound}$
  $[_{instruction}$ The first step consists in opening the computer box,$]$
  $[_{instruction}$ then to place it on a large, clean surface,
    $[_{elaboration\ or\ precaution}$ where you make sure there is no risk
      to damage electronic components,$]]$
  $[_{instruction}$ and then to withdraw all the caches at the PC front.$]]$

In the second example, an argument of type advice is introduced; it is composed of two instructions (later called conclusions) and a conjunction of three supports which motivate the two instructions.

$[_{instructional-compound}$
  $[_{goal}$ To clean leather armchairs,$]$
  $[_{argument:advice}$
    $[_{instruction}$ choose specialized products dedicated to furniture$]$,
    $[_{instruction}$ and prefer them colorless,$]$
    $[_{support}$ they will play a protection role, add beauty, and
      repair some small damages.$]]]$

Identifying rhetorical relations in this type of text is not straightforward. Some relations have a few marks associated whereas others are largely pragmatic and need some knowledge of the domain to be identified by a reader. We observed a few, partial, hierarchical relations between the items that build up an instructional compound. Scope priorities come in three groups. The first group is composed of goals and conditions, then, at a second level come causal, deontic and illucotionary elements operating over instructions. At the lower level, we have subordinated instructions, attached to the core instructions.

2.3 Implementation of instructional compounds

The actual schema for recognizing instructional compounds is quite simple at the moment, but results are quite satisfactory. Basically, such a compound contains at least one instruction. It is then delimited as follows:

- any element in an enumeration (typographically marked) is an instructional compound,
- in a paragraph which is not an enumeration, an instructional compound is delimited by expressions which induce an idea of strong break (even though this term is quite fuzzy). Such marks are for example: goal or conditional expression, end of paragraph, strong temporal mark (after two hours, when this is over, at the end of, and so on).

We have manually annotated 160 procedural texts. This is not an easy task due to the complexity of the structures at stake. Then these were compared with results obtained automatically. These will be used fully or in part to test the system. We selected texts we understand so that the risk of errors is limited as much as possible. This is presented in detail in (Delpech et al. 2008) where Kappa tests are realized to evaluate the homogeneity of human judgements.

For instructional compounds, for the three domains with best title and instruction recognition rate (do it yourself, cooking, and social life), we obtained the following results, based on a small corpus of data (60 texts):

Domain	Recall	Precision
cooking receipes	0.95	1.00
do it Yourself	0.89	0.98
social life	0.88	0.98

We have not yet attempted to implement an efficient system, but we are able to fully tag about 500 million web pages per hour, on a Pentium 4 3GhZ dual core machine with 4 Gigabyte RAM. This process includes cleaning web pages, running TreeTagger, and tagging titles, instructions and instructional compounds.

3 Identifying arguments in procedures

In this section let us first give a quite informal definition of what an argument is, and how it interacts with the goal-instructions structure. Let us then focus on warnings and advices which are, by far, the most frequently encountered structures. Most warnings and advices are included into instructional compounds.

3.1 Argumentation and Action theories

Roughly, argumentation is a process that allows speakers to construct statements for or against another statement called the conclusion. These former statements are called supports. The general form of an argument is: **Conclusion 'because' Support** (noted as $C\ because\ S$). In natural language, conclusions usually appear before the support, but they may also appear after, to stress the support. A conclusion may receive several supports, possibly of different natures (advices

and warnings): *don't add natural fertilizer, this may attract insects, which will damage your young plants*. Arguments may be more or less strong, they bear in general a certain weight, mostly induced from the words they contain or from their syntactic construction (Anscombre et al. 1981, Moeschler 1985, Amgoud et al. 2001). In natural contexts, this weight is somewhat vague.

In the case of procedural texts, the representation and the role of arguments in a text can be modeled roughly as follows. Let G be a goal which can be reached by the sequence of instructions $A_i,\ i \in [1,n]$, whatever their exact temporal structure is. Any instruction A_i is associated with a pair $(g_i,\ p_i)$ where g_i is the gain associated with A_i (there is a gain only in case where A_i is an advice, improving G) and p_i is the penalty in case where the user (partly of fully) fails to realize A_i.

A subset of A_i are interpreted as explicit arguments (A_i, is a conclusion) when they are explicitly paired with a support S_i that stresses the importance of A_i (*Carefully plug in your mother card vertically, otherwise you will damage the connectors*) or when an advice is given. Their general form is: $A_j\ because\ S_j$ (we use here the term 'because' which is more vague than the implication symbol used in formal argumentation, because natural language is not so radical). Supports S_k which are negatively oriented are warnings whereas those which are positively oriented are advices. Neutral supports simply introduce explanations which are not arguments. For the other instructions, the support is just implicit (do the action otherwise you will run into problems).

Similarly to the principles of argumentation theory, but within the framework of action theory (e.g. Davidson 2003), it is a priori possible to evaluate for a given realization of the instructions, the gains w.r.t. the goal G (when advices are followed, improving G) and the penalties (when actions are not well performed, with or without warnings). In an abstract model, we can assign each instruction a gain and a penalty, however in practice this is a little bit difficult. At the moment, let's say that gains are a priori null, except when we have an instruction of type advice which is realized, in that case the gain is greater or equal to 1. Penalties are numbers greater than 0, high penalties corresponding to very crucial instructions. If an instruction is correctly realized, penalty is 0, if there is a complete failure, penalty is the assigned number, which may be infinite if the instruction is absolutely crucial.

Given a certain realization by a user, the success of a goal G is the sum of the gains on the one hand, and the sum of the penalties on the other. Gains and penalties do not compensate each other but operate at different levels. Since any A_i is in fact realized successfully to a certain degree by the user, gains and penalties (which are values given a priori) need to be weighted, i.e. paired with a success measure, respectively μ and τ, each of these weights being included in $[0,1]$. Then, for a given execution of the goal G, we have:

$$\text{gain(G)} = \sum_{i=1}^{n} g_i \times \mu_i \qquad \text{penalty(G)} = \sum_{i=1}^{n} p_i \times \tau_i$$

As can be noted, our definitions include terms which are gradual: 'more difficult', 'easier', because in practice, failing to realize an instruction properly does not necessarily means that the goal cannot be reached, but the user will just be less successful, for various reasons. In the natural language expressions of conclusions (the A_j) as well as of supports, there are many modals or classes of verbs (like risk verbs) that modulate the consequences on G, contrast for example:

> *use professional products to clean your leathers, they will give them a brighter aspect.*

with:

> *carefully plug in your mother card vertically, otherwise you will most likely damage its connectors*.

In the latter case, the goal 'mounting your own PC' is likely to fail (the instruction at stake will be assigned a high penalty), whereas in the former, the goal 'cleaning your leathers' will just be less successful, but there is a gain g_i associated.

3.2 Processing arguments

From the above observations, we have investigated the different forms arguments may take and how they are realized in French. We noted that, in a very large number of cases, arguments in procedural texts can be identified by means of specific terms, i.e. there is no need to make complex parses or inferences. For most of them, they are embedded into instructional compounds, it is therefore quite easy to delimit them. Their scope is in general the compound, and their delimitation is quite simple. Most of the time, arguments are introduced by a goal or a cause connector. They are quite often either a complete, independent sentence following an instruction, or a subordinated clause ending a sentence. As a result, their recognition is relatively well portable from one procedural domain to another, with only mainly generic vocabulary involved.

We have defined a set of patterns that recognize instructions which are conclusions and their related supports. We defined those patterns from a development corpus of about 1,700 texts from various domains (cooking, do it yourself, gardening, video games, social advices, etc.). The study is made on French, English glosses are given here for ease of reading. The recognition problem is twofold: identifying propositions as conclusions or supports by means of specific linguistic marks and a few typographic marks, and then delimiting these elements. In general, boundaries are either sentences or, by

default, instructional compound boundaries. In procedural texts, roughly, the proportion of advices and warnings is almost equivalent.

Processing warnings

Warnings are basically organized around a unique structure composed of an 'avoid expression' combined with a proposition. The variations in the 'avoid expressions' capture the illocutionary force of the argument via several devices, ordered here by increasing force:

1. 'prevention verbs like avoid' NP / to VP (*avoid hot water*)
2. do not / never / ... VP(infinitive) ... (*never put this cloth in the sun*)
3. it is essential, vital, ... to never VP(infinitive).

In cases where the conclusion is relatively weak in terms of consequences, it may not have any specific mark, its recognition is then based on the observation that it is the instruction that immediately precedes an already identified support.

Supports are propositions which are identified from various marks: (a) via connectors such as: *otherwise, under the risk of,* etc., in French: *sinon, car, sous peine de, au risque de* or via verbs expressing consequence; (b) via negative expressions: *in order not to, in order to avoid, etc.*; (c) via specific verbs such as risk verbs introducing events (*you risk to break*) — in general the embedded verb has a negative polarity; (d) via the presence of very negative terms, such as: nouns: *death, disease, etc.*, adjectives, and some verbs and adverbs. We have a lexicon of ca. 200 negative terms found in our corpora.

Some supports have a more neutral formulation: they may be a portion of a sentence where a conclusion has been identified. For example, a proposition in the future tense or conditional following a conclusion is identified as a support. However, some supports may be empty, because they can easily be inferred by the reader. In that case, the argument is said to be truncated.

Patterns are implemented in Perl and are included into the global system (the TextCoop software). From the above observations, with some generalizations and the construction of lexicons of marks, we have summarized the extraction process in only eight patterns for supports and three patterns for conclusions. Arguments are tagged by XML tags. We carried out an indicative evaluation (e.g. to get improvement directions) on a corpus of 66 texts over various domains, containing 262 arguments. We get the following results for warnings: (supports well delimited are with respect to warnings correctly identified):

Conclusion recognition	**Support recognition**	**Conclusions well delimited**	**Supports well delimited**
88%	91%	95%	95%

Processing advices

Conclusions of type advice are identified essentially by means of two types of pattern (in French):

1. advice or preference expression followed by an instruction. The preference expression may be a verb or a more complex expression: *is advised to, prefer, it is better, preferable to, etc.*;

2. expression of optionality or of preference followed by an instruction: *our suggestions: ...,* or expression of optionality within the instruction (*use preferably a sharp knife*).

In addition, as for warnings, any instruction preceding a support of type advice is a conclusion. Supports of type advice are identified on the basis of 3 distinct types of pattern:

1. Goal exp + (adverb) + positively oriented term. Goal expressions are e.g.: *in order to, for*, whereas adverb includes: *better, more* (in French: *mieux, plus, davantage*), and positively oriented term includes: nouns (*savings, perfection, gain*, etc.), adjectives (*efficient, easy, useful*, etc.), or adverbs (*well, simply*, etc.). For this latter class of positively oriented terms we constructed a lexicon that contains about 50 terms.

2. Goal expressions with a positive consequence verb (*favor, encourage, save*, etc.), or a facilitation verb (*improve, optimize, facilitate, embellish, help, contribute*, etc.),

3. The goal expression in (1) and (2) above can be replaced by the verb 'to be' in the future: *it will be easier to locate your keys.*

A short example of an annotated text is given in Figure 1 below. Similarly as above, we carried out an indicative evaluation on the same corpus with 68 texts containing 240 manually identified advices (again, delimitation results are calculated with respect to advices correctly identified):

Conclusion recognition	**Support recognition**	**Conclusions well delimited**	**Supports well delimited**	**Support/Conclusion correctly related**
79%	84%	92%	91%	91%

As the reader may note, results are less satisfactory than for warnings. This is mainly due to the fact that advices are expressed in a much 'softer' way than warnings, with less emphasis and strength, therefore, terms typical of advices are not necessarily strongly marked, when present.

[$_{procedure}$
[$_{title}$ How to embellish your balcony
[$_{Prerequisites}$ 1 lattice, window boxes, etc.] ...
[$_{instructional-compound}$ In order to train a plant to grow up a wall, select first a sunny area, clean the floor and make sure it is flat...
[$_{Argument}$ [$_{Conclusion:Advice}$ You should better let a 10 cm interval between the wall and the lattice.]
[$_{Support:Advice}$ This space will allow the air to move around, which is beneficial for the health of your plant.] ...]]]]

Figure 1: An annotated procedure

The terms involved in advice as well as warning patterns are mostly domain independent, they are also quite limited in number. Their variations are mainly due to the author's style and the target audience. Finally, it seems that our extraction mechanism can be used to extract arguments in a large number of non-procedural texts such as news. This is very tentative but a few tests on French news indicates an accuracy of about 75%, but these also contain a few rewards and threats. We get really good results with teaching texts, which may be felt to be at the boarder line of procedural texts, but which contain quite a lot of reward expressions since interactions are more prominent (with the teacher). In terms of multilinguality, we are at the moment developing the same approach for Thai (at Kasetsart univ., Bangkok) applied to the treatment of rice.

[$_{instructional-compound}$ En decembre-janvier, effectuer la taille d'équilibrage et de nettoyage de vos arbres.
[$_{Argument}$ [$_{Conclusion:warning}$ La première année de fructification, éliminer tous les fruits au moment où ils se développent,]
[$_{Support:warning}$ Cela évite d'épuiser l'arbre.] ...]]

Figure 2: Annotated warning (gloss: *In December-January, make the first pruning and cleaning of your trees. The first year with fruits, eliminate all fruits, this avoids the tree to run out.*)

3.3 Dealing with empty supports

Considering do-it-yourself and gardening texts, we noted that about 2/3 of the arguments are not supported. This very large number of unsupported arguments, in such typically procedural texts, can be explained by several factors: (1) procedural texts are more oriented towards action than control, (2) some

[$instructional-compound$ [$goal$ Les étagères de votre meuble doivent être lisses.
[$Argument$ [$Conclusion{:}advice$ Utilisez si possible une ponceuse électrique,]
[$Support{:}advice$ Ce sera beaucoup plus rapide et vraiment moins fatigant.] ...]]

Figure 3: Anotated advice (gloss: *The shelves of your furniture must be very smooth. Use if possible an electric sander, this will be faster and less tiring.*)

supports, possibly complex, could in fact introduce doubts or confusions, (3) some explanations (supports) may be too complex to be understood by a casual user, and (4) supports are sometimes sufficiently explicit in the conclusions (*do not scatter seeds by high winds! = they won't go where you want them to go*). In socially-oriented procedural texts, supports are often much more explicit, but this may differ depending on the author's style.

We noted that realized supports correspond to two main trends: (1) supports that express general requirements such as: efficiency of actions, security, ease of execution, adequate execution, speed, aesthetics, lower cost, etc. and (2) supports that cover more precise, domain dependent situations (*avoid pruning trees when temperature drops below zero*).

Reconstructing empty support is still under research, let us note here the still very tentative directions we are investigating, which require different forms of inference. For empty supports corresponding to general requirements, we infer a generic support based on those requirements, e.g.: *mounting your computer: use a flat and clean surface.* induced support: 'for a better ease of execution'. From our observations (which need further confirmation), generic supports are in general triggered by adjectives or by general purpose verbs used in the conclusion.

The second situation (empty support in a domain dependent situation) is more delicate and requires domain or lexical knowledge. We are investigating the use of principles of the Generative Lexicon (Pustejovsky 1991) for that purpose. Very briefly, *wind* has in its telic role several predicates like *push, take away, scatter, disperse, break, damage, ...* When applied e.g. to gardening, such as planting new flowers, since these are not so mobile when planted, a predicate like *break* or *damage* can be selected (selection principles in the Generative lexicon remain however an open and very delicate problem). Then, from a statement such as: *avoid planting flowers by high winds* the support: *because wind will damage or break flowers* can be inferred.

4 Perspectives

The work presented here complements the tagging of titles and instructional compounds in procedural texts of various domains, as reported in (Delpech

et al. 2007, 2008). It allows us to have a quite comprehensive analysis of procedural texts, which turn out to have a very rich structure. Annotated corpus is available on request. We plan to include in our system the treatment of conditional expressions realized in (Bouffier et al. 2007), since we have not studied this phenomenon (and the associated scoping problems). We will then examine how illocutionary force is expressed. Finally, we plan to investigate the main lines of the facilitation structure.

We analyzed the forms arguments of type advice and warning may take, and have implemented and tested a system that tags those structures and attempts at reconstructing empty supports. At this level, there is still linguistic and formal work to be carried out, for example to evaluate the illocutionary force of arguments and to better settle this work within action theory. We believe we have a very useful corpus of examples of arguments, of much interest for research in argumentation theory. An important result is that arguments can be recognized on the basis of relatively simple parameterized patterns. The terms involved in those patterns are mostly domain independent. They may just vary depending on the author's style and the target audience. Finally, it seems that our extraction mechanism can be used to extract arguments in a large number of non-procedural texts such as news.

Besides studying the textual structure of procedural texts and responding to How-to questions from the analysis of these texts, a major application of this work is the construction of **domain know-how knowledge base**, quite basic but which could be subject to interesting generalizations. Obviously, to make this knowledge optimal, it would be useful to associate with every statement a formal representation that supports inference, data fusion, etc. This domain know-how knowledge base of advices, hints and warnings is of much importance for different types of users who have a procedure to realize a task but who want to know more before starting. Some psychological experiments have shown that, besides instructions given in procedural texts, users are very much interested in what remains implicit in those texts: what you are supposed to know or care about (but have no means to ask). Although there are already available, but manually constructed, such bases of advices, it is of much interest to construct it automatically, using the instructional compound as an advice unit.

Acknowledgements

This paper relates work realized within the French ANR project TextCoop. We thank its partners for stimulating discussions. We also thank Leila Amgoud and Daniel Kayser for very useful discussions on this work. Finally, we are very grateful to three reviewers for their detailed comments which helped improve and clarify this paper.

References

Adam, J.M., *Types de Textes ou genres de Discours? Comment Classer les Textes qui Disent De et Comment Faire*, Langages, 141, pp. 10–27, 2001.

Amgoud, L., Bonnefon, J.F., Prade, H., *An Argumentation-based Approach to Multiple Criteria Decision*, in 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU'2005, Barcelona, 2005.

Amgoud, L., Parsons, S., Maudet, N., *Arguments, Dialogue, and Negotiation*, in: 14th European Conference on Artificial Intelligence, Berlin, 2001.

Anscombre, J.-Cl. Ducrot, O., *Interrogation et Argumentation*, in Langue francaise, no 52, L'interrogation, 5 - 22, 1981.

Aouladomar, F., Saint-Dizier, P., *An Exploration of the Diversity of Natural Argumentation in Instructional Texts*, 5th International Workshop on Computational Models of Natural Argument, IJCAI, Edinburgh, 2005.

Bouffier, A., Poibeau, T., *Re-engineering free texts to obtain XML documents: a discourse based approach*, RANLP 2007.

Cruse, A., *Lexical Semantics*, Cambridge Univ. Press, 1986. Delin, J., Hartley, A., Paris, C., Scott, D., Vander Linden, K., *Expressing Procedural Relationships in Multilingual Instructions*, Proceedings of the Seventh International Workshop on Natural Language Generation, pp. 61-70, Maine, USA, 1994.

Davidson, D., *Actions, Reasons, and Causes*, Journal of Philosophy, 60, 1963

Delpech, E., Murguia, E., Saint-Dizier, P., *A Two-Level Strategy for Parsing Procedural Texts*, VSST07, Marrakech, October 2007.

Delpech, E., Saint-Dizier, P., Anonymous, Investigating the Structure of Procedural Texts for Answering How-to Questions, LREC 2008, Marrakech.

Gardent, C., Discourse tree adjoining grammars, report nb. 89, Univ. Saarlandes, Saarbrucken, 1997.

Kosseim, L., Lapalme, G., *Choosing Rhetorical Structures to Plan Instructional Texts*, Computational Intelligence, Blackwell, Boston, 2000.

Moeschler, J., *Argumentation et Conversation, Eléments pour une Analyse Pragmatique du Discours*, Hatier - Crédif, 1985.

Pustejovsky, J., *The Generative Lexicon*, Computational Linguistics, 17(4), 1991.

Rosner, D., Stede, M., *Customizing RST for the Automatic Production of Technical Manuals*, in R. Dale, E. Hovy, D. Rosner and O. Stock eds., Aspects of Automated Natural Language Generation, Lecture Notes in Artificial Intelligence, pp. 199–214, Springler-Verlag, 1992.

Takechi, M., Tokunaga, T., Matsumoto, Y., Tanaka, H., *Feature Selection in Categorizing Procedural Expressions*, The Sixth International Workshop on Information Retrieval with Asian Languages (IRAL2003), pp. 49–56, 2003.

Talmy, L., Towards a Cognitive Semantics, vol. 1 and 2, MIT Press, 2001.

Vander Linden, K., *Speaking of Actions Choosing Rhetorical Status and Grammatical Form in Instructional Text Generation* Thesis, University of Colorado, 1993.

Webber, B., D-LTAG: extending lexicalized TAGs to Discourse, Cognitive Science 28, pp. 751–779, Elsevier, 2004.

Yin, L., *Topic Analysis and Answering Procedural Questions*, Information Technology Research Institute Technical Report Series, ITRI-04-14, University of Brighton, UK, 2004.

Zuckerman, I., McConachy, R., Korb, K., Using Argumentation Strategies in Automatic Argument Generation, INLG 2000, Israel.