# Constructing a Know-How Repository of Advices and Warnings from Procedural Texts

Lionel Fontan
IRIT
118 route de Narbonne
31062 Toulouse cedex France
antonin_follet@hotmail.fr

Patrick Saint-Dizier
IRIT-CNRS
118, route de Narbonne
31062 Toulouse cedex France
stdizier@irit.fr

## ABSTRACT

In this paper, we show how a domain dependent know-how textual database of advices and warnings can be constructed from procedural texts. We show how arguments of type warnings and advices can be annotated and extracted from procedural texts, and propose a format and a strategy to automatically generate a know-how textual database.

## Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

## General Terms

Human factors, Experimentation

## Keywords

structure and content analysis, text semantics, automatically generated document

## 1. INTRODUCTION

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. 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. [2]. 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.

We have developed a quite detailed analysis of the textual structure of procedural texts, identifying their main basic components as well as their global structure. We defined two levels of processing: a segmentation level that basically tags structures considered as terminal structures (titles, instructions, advices, prerequisites, etc.) and a grammar that binds these terminal structures to give a global structure to procedural texts.

An important aspect of this project is to build a textual repository of advices and warnings related to an application domain, that reflects several forms of know-how on this domain. Such repositories exist, but they have been build completely manually, often in a wiki fashion. Our goal is then to allow users not only to query procedural texts via How to questions, but also to investigate how to model, create and access a repository of advices and warnings about a certain task, while keeping the context of the task.

We have already studied the instructional aspects of procedural texts and implemented a quite efficient prototype within the TextCoop project [2,4] that tags texts with dedicated XML tags. In this paper, after a brief categorization of explanation structure as found in our corpus of procedural texts, we focus on the argumentation structure via the recognition of warnings and advices. Then, we show how a textual repository of advices and warnings can be produced.

## 2. THE EXPLANATION STRUCTURE IN PROCEDURAL TEXTS

### 2.1 A global view of the explanation structure

We first constructed a quite large corpus (about 1700 texts) from several domains (basic: cooking, do it yourself, gardening, and complex: social relations, health) from a large number of web sites. From this corpus, we established a classification of the different forms explanations may take. The main structures are facilitation and argumentation structures. These structures are organized as follows:

- **facilitation structures**, which are rhetorical in essence [5, 8], correspond to *How to do X ?* questions, these include two subcategories:
  (1) user help, with: hints, evaluations and encouragements and
  (2) controls on instruction realization, with two cases: (2.1) controls on actions: guidance, focusing, expected result and elaboration and (2.2) controls on user interpretations: definitions, reformulations, illustrations and also elaborations.
- **argumentation structures**, corresponding to *why*


*DocEng'08,* September 16–19, 2008, San Paulo, Brazil.

*do X ?* questions. These have either:
(1) a positive orientation with the author involvement (promises) or not (advices and justifications) or
(2) a negative orientation with the author involvement (threats) or not (warnings).

## 2.2 From instructions to instructional compounds

In most types of texts, we do not find just sequences of simple instructions but much more complex compounds composed of clusters of instructions, that exhibit a number of semantic dependencies between each other, that we call **instructional compounds**. These 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.

An instructional compound has a relatively well organized discourse structure, composed of several layers, these are developed in (Fontan et al. 2008, under submission), let us just summarize these here:
(1) The goal and justification level, indicates motivations for doing actions that follow in the compound (e.g. *in your bedroom, you must clean regularly the curtains..*),
(2) The instruction kernel structure, which contains the main instructions,
(3) The deontic and illocutionary force structures: consist of marks that operate over instructions: (a) deontic: obligatory, optional, forbidden or impossible, alternates (or), (b) illocutionary and related aspects: stresses on actions: necessary, advised, recommended, to be avoided, etc. These marks are crucial to identify the weight of an argument,
(4) the temporal structure, quite basic here,
(5) the causal structure,
(6) The rhetorical structure [8, 5] whose goal is to enrich the kernel structure by means of a number of subordinated aspects (realized as propositions, possibly instructions) among which, most notably: enablement, motivation, circumstance, elaboration, instrument, precaution, manner. A group of relations of particular interest in this paper are arguments, developed hereafter.

An example of an instructional compound, using the square bracket notation, is:
[$_{instructional\ compound}$
[$_{Goal}$ 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.]]]

We have here an argument of type advice which is composed of 2 instructions (later called a conclusion) and a conjunction of three supports which motivate the 2 instructions.

# 3. IDENTIFYING ARGUMENTS IN PROCEDURES

## 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. The former statements are called supports. The general form of an argument is : **Conclusion 'because' Support** (noted as $C$ *because* $S$). A conclusion may receive several supports, possibly of different natures (advices and warnings). Arguments may be more or less strong, they bear in general a certain weight, induced from the words they contain [1, 6].

The representation and the role of arguments in a procedural 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. A subset of those instructions is interpreted as arguments where each instruction ($A_j$, viewed as a conclusion) is paired with a support $S_j$ that stresses the importance of $A_j$ ($<$ *conclusion* $>$ Carefully plug in your mother card vertically, $<$ */conclusion* $>$ $<$ *support* $>$ otherwise you will damage the connectors $<$ */support* $>$). Their general form is: $A_j$ *because* $S_j$. Supports $S_k$ which are negatively oriented are warnings whereas those which are positively oriented are advices. Neutral supports are explanations.

## 3.2 Processing arguments

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 1700 Web 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 and relevant linguistic marks (sometimes also a few typographic marks), and then delimiting these elements. In general, boundaries are either sentences or, by default, instructional compound boundaries.

### 3.2.1 Processing warnings

Warnings are basically organized around an 'avoid expression' combined with a proposition. The variations around the 'avoid expression' capture the illocutionary force of the argument, 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).

Supports convey negative statements, they are identified from various marks:
(1) via connectors such as: *sinon, car, sous peine de, au risque de* (otherwise, under the risk of), etc. or via verbs expressing consequence,
(2) via negative expressions of the form: *in order not to, in order to avoid, etc.*
(3) via specific verbs such as risk verbs introducing an event (*you risk to break*). In general the embedded verb has a negative polarity.
(4) via the presence of very negative terms, such as: nouns: *death, disease, etc.*, adjectives, and some verbs and adverbs. We have a lexicon of about 200 negative terms found in our corpora.

Some supports may be empty, because they can easily be inferred by the reader. In that case, the argument is said to be truncated.

Linguistic marks have been optimized to produce patterns. These are implemented in Perl and are included into

the TextCoop software [4]. With some generalizations and the construction of lexicons of marks, we have summarized the extraction process in only 8 patterns for supports and 3 patterns for conclusions. The system is based on the linear execution of patterns, which possibly include automata. In procedural texts, 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. Those texts where manually annotated by a trained linguist, and the results were then compared with the system output. We get the following results for warnings:

| conclusion recognition | support recognition | (3) | (4) |
|---|---|---|---|
| 88% | 91% | 95% | 95% |

(3) conclusions well delimited (4) supports well delimited, with respect to warnings correctly identified.

#### 3.2.2 Processing Advices

Conclusions of type advice are identified essentially by means of two types of patterns (in French):
(1) advice or preference expressions followed by an instruction. The expressions may be a verb or a more complex expression: *it 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*).

Supports of type advice are identified on the basis of 3 distinct types of patterns:
(1) Goal exp + (adverb) + positively oriented term. Goal expressions are e.g.: in order to, for, whereas adverb includes: better (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.). We constructed a lexicon of positively oriented terms that contains about 50 terms.
(2) Goal expression 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 is given in Fig. 1 below.

Similarly as above, we carried out an indicative evaluation on the same corpus of 66 texts containing 240 manually identified advices. We get the following results for advices:

| conclusion recognition | support recognition | (3) | (4) | (5) |
|---|---|---|---|---|
| 79% | 84% | 92% | 91% | 91% |

(3) conclusions well delimited, (4) supports well delimited, both with respect to advices correctly identified. (5) support and conclusion correctly related.

## 4. CONSTRUCTING AND QUERYING A KNOW-HOW TEXTUAL DATABASE

A major application of this work is the construction of a **domain dependent know-how textual database**, which is probably quite basic, but which could be subject to interesting generalizations. 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 certain task but who need more support without having to go through dozens of web pages. Some psychological experiments have in fact shown that, besides instructions, users are very much interested in what remains implicit in those texts: what you are supposed to know or care about. This know-how database is aimed to fill in this kind of gap.

The work presented hereafter is still exploratory: we need to elaborate different ways of producing such a database, considering the type of questions users may have and the way they would like to access textual databases.

### 4.1 Constructing a text database of domain know-how

There are repositories of advices organized by sector of activity available on the Web (e.g. http://www.conseils-gratuit.com). These are constructed manually: most of these advices come from hints sent by readers of these pages. These repositories contain in general simple advices and also small procedures which are hints to better realize a certain task.

Texts have first to be processed as follows:
(1) cleaning web pages from irrelevant data (adds, forums, summaries, links, etc.),
(2) XML tagging the instructional aspects, with dedicated tags: tagging titles, and tagging instructional compounds and prerequisites [4], and
(3) tagging within instructional compounds advices and warnings based on the patterns given above.
Let us first present the construction of the domain know-how textual database of advices and warnings.

Then, the first level of structure of the database are domains. So far, domains are defined on a coarse-grained level by ourselves, they correspond to major classes in 'practical life' journals. Texts are classified into these domains manually, considering the origin of the web page (e.g. cooking site, do it yourself shops). We therefore process texts by domain, according to our corpus (about 8000 texts). In the database, the first level of structure are therefore domains: `house, cooking, administration, health, garden, computer, do it yourself, animals, beauty, society`.

Next, in the textual database it is necessary to have means to settle the context in which advices and warnings are uttered. For that purpose, we are experimenting the reference to text titles, which form the main navigation units users can manipulate. Therefore, below each of the domain top nodes, we have a list of items that correspond to procedures main titles (e.g. *boucher un trou avec du platre (fill up a hole with plaster))*. Since, for most domains we have several hundreds of documents, we need to organize those titles and abstract over them. This is organized around two axis:
(1) task oriented: where action verbs are grouped on the basis of closely related terms to form a single title (for that purpose we use our verb lexical base). A second level of generalization is carried out via several types of linguistic operations such as: skipping adjuncts and generalizing over the verb title via synonyms. In the end, we have **generalized titles** like: 'repairing walls' independently of the material or the technique used, e.g. with plaster. mastic, cement.
(2) object oriented: where we only keep track of the objects, viewed as a theme: wall, wood, plaster, etc. so that the user

< *procedure* > < *title* > How to embellish your balcony < /*title* >
< *Prerequisites* > 1 lattice, window boxes, etc.< /*prerequisites* >
....
< *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 att* = "*Advice*" > You should better let a 10 cm interval between the wall and the lattice. < /*Conclusion* >
< *Support att* = "*Advice*" > This space will allow the air to move around, which is beneficial for the health of your plant. < /*Support* >< /*Argument* > ... < /*instructional* − *compound* > .....
..... < /*procedure* >

**Figure 1: An annotated procedure**

< *domain* > do-it-yourself
< *topic* > topic: repairing walls
< *title* > repairing your walls with plaster< /*title* > < *support* > list of extracted support-conclusion pairs < /*support* >
< *title* > filling up holes in your walls< /*title* > < *support* > list of extracted support-conclusion pairs < /*support* >
..... < /*topic* >
< *topic* > painting walls .... < /*topic* >
.....</domain>.....
< *object* > plaster, < *title* > reparing walls < /*title* > lis of support-conclusions ....
cement, ..... < /*object* >.

**Figure 2: A sample of the know-how textual database**

can access the different operations these objects may undergo. Text titles appear below these objects to facilitate navigation.
These revised titles form a second level in the structure of the know-how textual knowledge base called 'topic'.

Below these two levels, we have the list of titles associated with advices and warnings. Fully expanded titles are used to make the procedure context more precise so that the scope of supports is more clear. In our experiment, the text units that we have access to are instructional compounds, which correspond to the various advice and warning forms found in manually realized repositories. However, compounds being inserted into a larger procedure may be somewhat elliptical in some cases. A short example is given in Fig. 2.

## 4.2 Querying the know-how textual database

In general, attempting to match queries directly with supports in order to get the advice, i.e. the associated conclusion does not lead to the best results because supports are often incomplete or they contain a lot of pronominal references. Our matching procedure therefore includes the taking into account of the page title and subtitles together with support contents. It seems that this leads to better results in terms of accuracy and relevance.

Related to Fig. 2, a query could be: *how to get smooth plaster surfaces on a wall ?*. Answering this question is realized via the following steps:
(1) based on keywords which appear as objects in the query, select a domain and a topic in the textual database.
(2) then, over the topics selected, match the query with one or more supports. Matching is obviously not direct and requires, as in most systems, some flexibility. Of interest here are adjectives, which abound, for which we developed scales [3] that capture the different language expressions of the properties they characterize.
(3) then supports associated with their conclusion are submitted to the user, based on a heuristics that orders them by decreasing likelihood, based on the matching quality. There are several well-known algorithms which can be used, among which the best match algorithm. In case there are too many responses, some additional devices can be integrated, like navigational tools.

**Acknowledgements** This work is funded by the French ANR programme, RNTL section. This is part of the TextCoop project. We thanks in particular Leila Amgoud and Daniel Kayser for constructive discussions about argumentation. We also thanks project members with whom we had in depth discussions.

## 5. REFERENCES

[1] Amgoud, L., Parsons, S., Maudet, N., *Arguments, Dialogue, and Negotiation*, in: 14th European Conference on Artificial Intelligence, Berlin, 2001.
[2] Aouladomar, F., Saint-Dizier, P., *Towards Answering Procedural Questions*, Workshop KRAQ05, IJCAI05, Edinburgh, 2005.
[3] Cruse, A., lexical Semantics, CUP, 1986.
[4] Delpech, E., Saint-Dizier, P., Investigating the Structure of Procedural Texts for Answering How-to Questions, LREC 2008, Marrakech.
[5] Kosseim, L., Lapalme, G., *Choosing Rhetorical Structures to Plan Instructional Texts*, Computational Intelligence, B. Blackwell, Boston, 2000.
[6] Moschler, J., *Argumentation et Conversation*, Hatier - Crédif, 1985.
[7] Talmy, L., Towards a Cognitive Semantics, vol. 1 and 2, MIT Press, 2001.
[8] Vander Linden, K., *Speaking of Actions Choosing Rhetorical Status and Grammatical Form in Instructional Text Generation* Thesis, University of Colorado, 1993.