Investigating the Structure of Procedural Texts for Answering How-to Questions

Estelle Delpech, Patrick Saint-Dizier

IRIT-CNRS
118 route de Narbonne
31062 Toulouse cedex France
delpech_estelle@yahoo.fr, stdizier@irit.fr

Abstract

This paper presents ongoing work dedicated to parsing the textual structure of procedural texts. We propose here a model for the intructional structure and criteria to identify its main components: titles, instructions, warnings and prerequisites. The main aim of this project, besides a contribution to text processing, is to be able to answer procedural questions (How-to? questions), where the answer is a well-formed portion of a text, not a small set of words as for factoid questions.

1. Situation and Aims

The main goal of this work is to be able to answer procedural questions, which are questions whose induced response is typically a fragment, more or less large, of a procedure, i.e., a set of coherent instructions designed to reach a goal. Recent informal observations from queries to Web search engines show that procedural questions is the second largest set of queries after factoid questions (de Rijke, 2005). In this paper, we focus on the analysis of procedural structures in texts (titles, instructions, warnings, prerequisites, etc.).

Answering procedural questions thus requires to be able to extract not simply a word in a text fragment, as for factoid questions, but a well-formed text structure which may be quite large. Analysing a procedural text requires a dedicated discourse analysis, e.g. by means of a grammar. Such grammars are not very common yet due to the complex intertwinning of lexical, syntactic, semantic and pragmatic factors they require to get a correct analysis. Producing responses which are well-formed text portions is not proper to procedural questions. Many other types of questions require texts as responses: why questions, but also evaluative or comparative questions. Next, any kind of cooperative answering framework requires the production of informational elements such as explanations, examples or arguments which are basically textual and strongly organized.

Procedural texts are organized sets of instructions, they may also be sets of advices, as in social behavior texts. 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. Even if procedural texts adhere more or less to a number of structural criteria, which may depend on the author's writing abilities and on traditions associated with a given domain, we observed a very large variety of realisations, which makes parsing such texts quite challenging.

Procedural texts explain how to realize a certain goal by means of actions which may be temporally organized. Procedural texts can indeed be a simple, ordered list of instructions to reach a goal, but they can also be less linear, outlining different ways to realize something, with arguments, advices, conditions, hypothesis, preferences. They also often contain a number of recommendations, warnings, and comments of various sorts. The organization of a procedural text is in general made visible by means of linguistic and typographic marks.

Research on procedural texts was initiated by works in psychology, cognitive ergonomics, and didactics, (Mortara et ali. 1988) (Adam 1987), (Greimas 1983), (Kosseim 2000) to cite just a few. Several facets, such as temporal and argumentative structures have then been subject to general purpose investigations in linguistics, but they need to be customized to this type of text. There is however very little work done in Computational Linguistics circles. The present work is based on a preliminary experiment we carried out (Delpech et ali., 07), (Aouladomar, 05) where a preliminary structure was proposed, from corpus analysis.

In this paper, we summarize our results, focussing (1) on the conceptual notion of intructional compounds, which does capture the complexity just advocated, (2) on the recognition of titles, instructions and instructional compounds and (3) on the modelling and implementation of a simple text grammar system that accounts for the overall text structure w.r.t. to procedurality. A quite comprehensive evaluation was carried out that we report here. This work is part of the French ANR-RNTL TextCoop project.

2. The structure of procedural texts: Instructional Compounds

The main construction of procedural texts is the goal-plan structure. They may show a hierarchical structure composed of subgoals. This constitutes the skeleton of a procedural text. Procedural texts therefore contain two basic structures: titles, interpreted as goals (on which question matching procedures will apply), and instructions serving these goals. However, in most types of texts, we do not have just sequences of simple instructions but much more complex compounds composed of clusters of instructions. We noted that 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. Procedural texts also

contain general purpose prerequisites and warnings, besides those included in instructional compounds. All these elements are, in fact, essential to a good understanding of procedural texts: for example, explanations and arguments help the user understand why an instruction must be realized and what are the risks if he does not do it properly.

Let us essentially, in this contribution, focus on the instructional compound structure, which is, by far, the most complex one. It has a relatively well organized discourse structure, composed of several layers, which are:

- The **goal and justification**, which has wider scope over the remainder of the compound, indicates motivations for doing actions that follow in the compound (e.g. *in your bedroom, you must clean regularly the curtains...*, which here motivates actions to undertake). It gives the funtamental motivation of the compound.

- The **instruction kernel structure**, which contains the main instructions. These can be organized temporally or be just sets of actions. 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. 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, introduced below.

- The **deontic and illocutionary force structures**: consist of marks that operate over instructions, outlining different parameters:

  - deontic: obligatory, optional, forbidden or impossible, alternates (or),

  - illocutionary and related aspects: stresses on actions: necessary, advised, recommended, to be avoided, etc.

- a **temporal structure** that organizes sequences of instructions (and at a higher level instructional compounds). In general, the temporal structure is very simple, with sequences of actions to carry out. In some case, parallel actions are specified, which partially overlap.

- The **conditional structure**: introduces conditions over instructions within the compound or even over the whole instructional compound. We encounter quite a lot of structures identifying mutually exclusive cases.

- the **causal structure** that indicates the goal of an action. We identify four types of causal relations, following (Talmy 2001): intend-to (direct objective of an action 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*).

- The **rhetorical structure** 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: causality, enablement, motivation, argument for, advice, circumstance, elaboration, instrument, precaution, manner. A group of relations of particular interest are arguments (positive or negative). The rhetorical structure is in general composed of instructions (satellites) related to the kernel instructions.

Let us now give an illustrative example (translated from French), extracted from the 'Do-It-Yourself Home' domain: *In the bedroom, it is necessary to clean curtains. These are cleaned first with a vacuum-cleaner to remove dust, then, if they are in cotton, they can be washed in the washing machine at 60 degrees; if they are white, it is even recommended to add some bleech so that they look whiter. With some starch, they can be easily ironed.*

In this text, the sequence: *In the bedroom, it is necessary to clean curtains* is analyzed as a justification of the actions to undertake. The next portion: *These are cleaned first with a vacuum-cleaner to remove dust, then, if they are in cotton, they can be washed in the washing machine at 60 degrees.* is the instruction kernel, where the last instruction is associated with a condition. Finally, *If they are white, it is even recommended to add some bleech so that they look whiter. With some starch, they can be easily ironed.* are two subordinated clauses, analyzed as being in the rhetorical relation advice to the kernel instructions.

Another example in parenthetical format (French gloss) is the following:

[The first step consists in opening the computer box, then to place it on a large, clean surface,

[$_{argument}$ where you have ample space to work comfortably,]

and then to withdraw all the cashes at the PC front.]

A more complex case is:

[[$_{Goal}$ To clean leathers,]

[$_{instruction}$ choose specialized products dedicated to furniture,

[$_{advice}$ [$_{instruction}$ and prefer them colorless],

[$_{arguments}$ 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. One of our goals is to focus on the explanation - argumentation structure and to propose a model which could also be used for response generation.

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 illocotionary elements operating over instructions. At the lower level, we have subordinated instructions, attached to kernel instructions.

3. Recognizing Titles, Instructions and Instructional Compounds

Let us now describe in some detail how the different structures that compose procedural texts are recognized. The work is realized in two steps. First, a segmenter identifies basic text elements such as titles, instructions, etc. These form the terminal elements. Then a grammar is applied on those terminal elements to bind them into a text structure. We call it a *Text Grammar*.

3.1. Cleaning Web texts

The input of our system are raw Web pages. To be able to correctly tag the procedural elements of these texts, it is necessary to eliminate useless information (advertising, summaries, links to blogs, comments, etc.). This useless information can represent up to 66% of the text. To carry this out, we need

1. to extract relevant text, that is, any kind of text that is not navigation help, advertisements or comments posted by cybernauts and

2. to select and to simplify the html tags in so as to keep the main typo-dispositional information (paragraph breaks, subdivisions of paragraphs into lines, lists and their subdivision into elements, emphasis).

Although (2) was quite an easy task, we had some difficulties achieving (1). We designed an algorithm that returns, for each paragraph, if its contents can be considered as relevant or not. It mainly uses paragraph length and proportion of closed-class words criteria. We evaluated it on 100 Web pages, from 12 different web sites. The results compared to a manual treatment are quite good, we have 0,95 precision and 0,76 recall.

When the text is 'clean', we apply the Treetagger on it to identify its morpho-syntactic terms. We just keep some categories of interest to us (e.g. verbs, connectors). We also make some revisions since, in French, the imperative form, which is central to our system of extraction patterns, is often identified as present indicative tense.

3.2. Recognizing Titles

For answering How-to questions it is obviously of much importance to recognize titles, which, in fact, mostly express goals of various levels. A second challenge is to possibly identify title hierarchies in complex or long texts. Automatically identifying titles is quite challenging and has been seldom addressed in the past. Obviously criteria depends on the type of text (pdf, word, html, etc.), the quality of the encoding, the type of text (procedural, roman, news, etc.) and the domain at stake. Let us concentrate here on procedural texts, encoded in html format, from various sources, styles and domains. The next problem for us is that a number of titles in web pages are irrelevant with respect to the procedure at stake, they are rather advertising, web services ('click here for more') or summaries, to cite just a few. Besides recognizing titles as such, our task is also to be able to concentrate on titles related to a procedure, so that these can be used for answering questions.

Titles are short text sequences, highlighted (bold, color, underlined, size or type of font, etc.). A first observation is that html encodings are, by far, not homogeneous. Titles are coded with the tag $< h_n >$ in only 20% of the cases over the 600 titles observed. In most cases, the tag $< b >$ is used, possibly also $< emp >$, $< u >$ and a few others (macros...). Low level titles even have more unexpected encodings. Encodings may be quite homogeneous within a given web site, but heterogeneity prevails over different sites, even in the same domain.

To be more precise, we observed that, roughly:

- 80% of titles are encoded with $< b >$

- 57% of the total of $< b >$ used in texts encode titles

- 64% of the total of $< h >$ used in texts encode titles.

This means that we need to consider additional criteria, among which:

- typography (spacing w.r.t. paragraphs before and after,

- the contents (number of words, inflected verbs) of the segment assumed to be a title,

- the type of elements after the title (e.g. instructions, which are a good indicator of a procedural title).

Titles are identified in two steps. First, an algorithm traverses paragraphs of a text one by one, and assigns them one of the following tags: title, text or ambiguous. This first step is quite straightforward. From our investigations on procedural texts, a title is a paragraph composed of a unique sequence of words, less than 12 words long and bearing emphasis. The tag text will be assigned without any doubt if the paragaph is subdivided into smaller units or is longer than 12 words. Ambiguous parapraphs are mainly short sequences of words (12 words or less) with no emphasis.

The second step disambiguates the ambiguous paragraphs one by one, using the tags assigned by the first step to their surrounding paragraphs. For example, an ambiguous paragraph between two paragraphs tagged as text will be considered as a text. Similarly, we have the following rules:
'an ambiguous paragraph between two titles is a text',
'an ambiguous paragraph followed by a title becomes a text'
'an ambigous paragraph becomes a title if it is the first paragraph of the text', etc.

This second step also operates some repairs on the tags yielded by the first step. For example, any sequence of more than two titles, i.e : "title title title", will be changed to "title title text".

The title hierarchy is very difficult to identify without content analysis. In fact, it is often largely pragmatic in nature. For example in 'The pizza Margarita .... the paste .... the topings .... the serving ...'. It is impossible a priori to hierarchically organize those titles if you do not know what pizzas look like.

However, standard procedural texts are not very long and tend to be relatively linear. This means that, besides the page title, we observed in 80% of our texts not more than 2

levels of titles (exluding the main title). We observed two regular types of titles that can be correlated to some form of hierarchy. Type 1 is a title separated from the paragraph that follows by a $<p>$ tag. Type 2 is a title separated from the paragraph that follows by a $
$ tag. Although we still have no means to tell the exact level for titles, we can quite confidently say that a type 2 title will be at a lower level than a type 1 title, whatever the website or the domain. This information may be useful for question-title matching : type 2 titles are expected to introduce paragraphs that deal with more specific aspects of a procedure than paragraphs introduced by a type 1 title. Type 2 titles could help answering specific questions. One remaining difficulty for question-title matching is that titles have often a very elliptic structure.

3.3. Recognizing instructions and instructional compounds

3.3.1. Patterns for instructions

We noted that what is usually called an instruction ranges from clearly injunctive clauses to implicit prescriptions (this complexity is reflected in the complexity of manual annotation tasks, as reported in the evaluation section). Instructions are recognized on the basis of two factors: contents, around action verbs in certain forms to identify an instruction, and typographic or linguistic factors for its delimitation (beginning and end) via html tags, punctuation marks or connectors. Currently, we use a set of only 14 lexico-morphological patterns, that encompass the most prototypical ways of expressing instructions. The lexical resources needed are most notably: action verbs, incentive verbs, and related nouns and adjectives. They are generic and are, for a large part, reusable from one domain to another.

Most instructions can be recognized on the basis of patterns. Verb forms observed may not necessarily be only protypical of instructions. We keep here those which have the most injunctive weight. The most prototypical ones include verbs in the following forms, for French:

1. infinitive forms (typical of e.g. do-it-yourself, video games solutions),

2. infinitive forms in independent propositions (typical e.g. of coocking receipes),

3. modal constructions (you must, it is necessary to...) followed by an infinitive form, and other types of expressions with a modal value,

4. impersonal expressions using the dummy pronoun 'on' (it) followed by an action verb,

5. the use of the modal 'pouvoir' (can), which is very recurrent, in particular in social and health contexts.

These structures cover in Frenck about 98% of the cases. They are therefore strongly prototypical. A few passive voice expressions, gerundives and pronominal expressions have been observed, but they are too ambiguous between instructional or not.

The recognizer (also called the segmenter) contains at the moment 14 patterns. It is implemented in standard Perl.

We plan to extend it further by specializing the patterns according to application domains. For example, and in order to improve recognition in domains like health and society we need to include constructions in French based, e.g. on the semi-auxiliary 'faire', some finite forms with action verbs, constructions using aspectual verbs (start by opening...) and a few fixed forms (this consists in opening...). In fact, this kind of extension is about the only major task to transpose our work to other procedural domains. We tested the initial set of patterns on a new domain: pedagogical texts, and results are comparable to those obtained for the do-it-yourself domain.

3.3.2. Recognizing 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) forms 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 e.g.: goal or conditional expression, end of paragraph, strong temporal mark (after two hours, when this is over, at the end of, etc.).
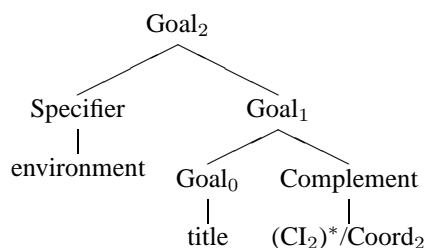
4. A text grammar

Finally, the different constituents presented above are tied by means of a 'text' grammar. This grammar is specifically dedicated to procedural texts, it is not as generic as those developed by e.g. (Webber 2004) for LDTags or (Gardent, 1997). It is based on X-bar syntax, that we have transposed to the discourse level. We just shwo here the main trees, for an easy reading.
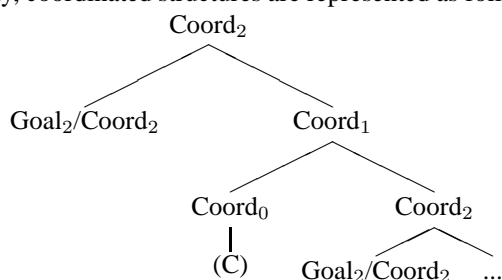
The highest level is Proc2:
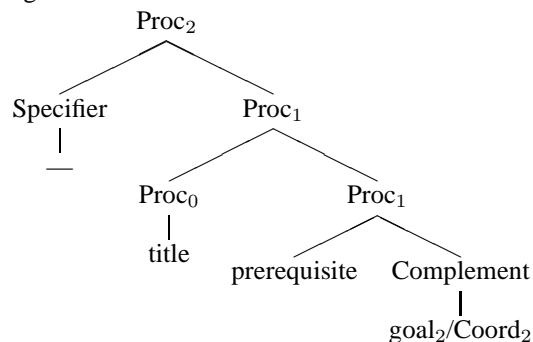
Then, a Goal2, corresponds to a title. The tree is the following, where embedded structures are allowed, since we have titles at different levels:

```
                    Goal_2
                   /      \
            Specifier      Goal_1
                |         /      \
          environment  Goal_0    Complement
                         |           |
                       title    (CI_2)*/Coord_2
```

CI = instructional compound.

Finally, coordinated structures are represented as follows:

```
                  Coord_2
                 /        \
      Goal_2/Coord_2       Coord_1
                          /       \
                    Coord_0        Coord_2
                       |          /      \
                      (C)   Goal_2/Coord_2  ...
```

The grammar runs in Prolog in our prototype. The output is an XML file that reflects the text structure.

5.   Evaluation

The evaluation we have carried out allows us to have an estimate of the overall quality and accuracy of the recognition mechanisms, outlining problems and gaps for future evolutions. From that point of view, it is an indicative evaluation.

5.1.   Annotation tasks for evaluation

The evaluation corpus is composed of 78 Web pages over 5 domains: coocking recipes, do-it-yourself, video game solutions, social life, and medical recommendations. The total number of words is 61159, this not very large, but we feel sufficient for an indicative evaluation, giving us directions to improve the system. The annotation unit is a sequence (a sentence, an isolated text fragment or an element of an enumeration, as specified in the instruction recognizer). For each sequence, annotators had to decide whether it is a title, an instruction (with the possibility to give certainty of judgement) or none of them. The corpus contains 4560 sequences, among which 511 titles and 1641 sentences containing at least one instruction.

The total work took about 15 hours of manual work. Decisions were quite often difficult to make for some types of texts where quite a lot of knowledge of the domain is required, as for video games. We tried to set up criteria so that instructions could be annotated in the most systematic and stable way possible. Then the two annotators had discussions (about 5 hours) to reach a consensus and propose a unique annotation for each text.

5.2.   Results for Instructions

The result was then compared to the annotations realized by the programme. Our strategy was in general to favor precision over recall, since even if some instructions are not recognized here and there, the question-answering system can still respond accurately. For instructions recognition we have the following results:

domain	recall	precision	certainty	kappa
coocking receipes	0.81	1	0.82	0.88
do it Yourself	0.77	0.95	0.84	0.76
social life	0.63	0.94	0.78	0.58
video games	0.38	0.96	0.58	0.45
medical notices	0.33	0.95	0.57	0.60

Fig. 1 - Recognition rates for instructions

Although the precision rate keeps high throughout the domains, the recall rate drops in parallel with the certainty score. This stems from the fact that only the most prototypical lexico-morphological patterns for instructions were implemented. Less prototypical patterns may be more ambiguous between instructional and not instructional. For example the use of passive voice is used, in procedural texts, for two main intents: to give instructions and to make descriptions. In our first implementation of the recognizer, we did not to implement it, deliberately favoring precision over recall.

5.3.   Results for Titles

Regarding titles, the results are slightly better than instructions, but also display irregularity over domains. This could be a hint as to the possible diversity of title typographical representation and complexity over domains, but the number of evaluated units (511) may not be high enough to conclude:

domain	recall	precision	certainty
coocking receipes	0.72	1	0.83
do it Yourself	0.8	0.96	0.87
social life	0.69	0.97	0.80
video games	0.61	0.93	0.74
medical notices	0.58	0.81	0.67

Fig. 2 - Recognition rates for titles

5.4.   Results for Instructional Compounds

Finally, for instructional compounds, for the three best domains, and with respect to the results obtained in each of these domains, we have the following results, based on a small corpus of data, due to the complexity of the manual analysis:

Instructional compound recognition:

domain	recall	precision
coocking receipes	0.95	1
do it Yourself	0.89	0.98
social life	0.88	0.98

Fig. 3 - Recognition rates for instructional compounds

We have not tried at this level to implement an efficient system, however, we can fully parse 500 Mo of web pages per hour, on a pentium4 3GhZ machine with 4 Go RAM. It should be quite easy to enhance efficiency to reach almost real-time performance needed for on-line question-answering.

6.   A Few Perspectives

The linguistic structure of texts and the methods to recognize titles, instructions and instructional compounds and the global text structure seem to be on the right track. We obviously need to deepen evaluation for whole texts, but

this is much more difficult due to the complexity of annotations.

To improve the domains with low level instruction recognition results, one direction would be to design domain dedicated recognizers, with specific patterns. Some more efforts are also necessary in large texts to identify title hierarchies. At the moment, we do not see any simple solution which does not involve heavy pragmatic or domain factors. An interesting feature is that, although we need quite a lot of resources, they are all the same for most domains and styles, making the analysis quite portable.

The next step of the project is to explore how How-to questions can match with titles (goals), and what kind of results must be returned to the user (the instructions below the title, more data containing prerequisites, several documents, etc.). Another area is the exploration of the structure of explanations and arguments in procedural texts which is a very important aspect, that supports, in fact, the instructions and that helps the user understand why he/she should do such or such task, and what are the consequences (warnings) if this is not done as required. In addition, advices are given to guide the reader, and help him to imporve the quality of the task or his efficiency.

7. 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.

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.

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.

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

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

Greimas, A., *La Soupe au Pistou ou la Conservation d'un Objet de Valeur*, in Du sens II, Seuil, Paris, 1983.

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

Luc, C., Mojahid, M., Virbel, J., Garcia-Debanc, C., Pery-Woodley, M-P., *A Linguistic Approach to Some Parameters of Layout: A study of enumerations*, In R. Power and D. Scott (Eds.), Using Layout for the Generation, Understanding or Retrieval of Documents, AAAI 1999 Fall Symposium, pp. 20-29, 1999.

De Rijke, M., *Question Answering: What's Next?*, the Sixth International Workshop on Computational Semantics, Tilburg, 2005.

Maybury, M., *New Directions in Question Answering*, The MIT Press, Menlo Park, 2004.

Moldovan, D., Harabagiu, S., Pasca, M., Milhacea, R., Goodrum, R., Grju, R., Rus, V., *The Structure and Performance of an Open-Domain Question Answering System*, Proceedings of the 38th Meeting of the Association for Computational Linguistics (ACL), Hong Kong, 2000.

Mortara Garavelli, B., *Tipologia dei Testi*, in G. Hodus et al.: lexicon der romanistischen Linguistik, vol. IV, Tubingen, Niemeyer, 1988.

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.

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.