

Some Challenges of Advanced Question-Answering: an Experiment with How-to Questions *

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

Abstract. This paper is a contribution to text semantics processing and its application to advanced question-answering where a significant portion of a well-formed text is required as a response. We focus on procedural texts of various domains, and show how titles, instructions, instructional compounds and arguments can be extracted.

Keywords: text semantics, question answering.

1. Introduction

Question answering (QA) is an area that operates on top of search engines (such as Google, Exalead, Yahoo, etc.) in order to provide users with more accurate and elaborated responses where search engine response outputs remain punctual, difficult to understand, and sometimes incoherent. QA builds on top of search engine responses via language processing tools, reasoning mechanisms and response production techniques. QA has been recognized as an essential component of man-machine communication since it greatly improves communication with the Web or textual databases, allowing users to communicate in their own language, in order to get much more precise, accurate and user-tailored responses.

Factoid questions (questions about facts, e.g.: *what is the capital of X ?*) have been largely investigated (witness, e.g. the TREC competition), but the other types of questions, frequently encountered such as : procedural (how to), causal (why), evaluative and comparative questions (*what is the cheapest air ticket to go to Cebu from Singapore ?*) have received little attention so far. However, they correspond to the majority of the 'real' questions asked either by the large public or by professionals to the web or to textual databases. Questions cover a large number of everyday life (health, employment, administrative life, tourism, etc.) as well as technical areas (investments, competitive intelligence, etc.).

Complex questions require language processing (to process the question and on the outputs of search engines), reasoning (e.g. data fusion, incoherence solving, summarizing, etc.) and response production in natural language (in a way accessible and user-friendly). These complex questions are in fact those users ask most of the time, these users being professionals or from the large public. Complex questions are traditionally classified according to the type of response which is induced, e.g.:

- definition questions,

* Acknowledgments: we are grateful to a number of MA students who contributed to this work, among which: Estelle DELPECH, Lionel FONTAN, Isabelle DAUTRICHE and Clementine ADAM. We also thank the ANR-RNTL French program for supporting part of this research.

- how-to questions for procedures,
- why questions for causes/consequence (possibly involving chains of events),
- evaluative and comparative questions (asking for comparisons between two or more elements such as fares).

Considering the technology which is required, it is clear that search engines will not be able in the near future or even in the mid-term future to correctly process these questions due to the complexity of the task, the need of some domain knowledge and the necessity to provide natural language responses, beyond text extracts.

The current trend in TREC Question Answering is towards the processing of large volumes of open-domain texts (e.g. documents extracted from the World Wide Web). Open domain QA is a hard task because no restriction is imposed either on the question type or on the user's vocabulary. This is why most of the efforts (in evaluation campaigns such as TREC) are focused on answering factoid style questions, and, to a little extent, definition questions, using shallow text processing which is roughly based on pattern extraction or information retrieval techniques. However, QA should integrate more complex techniques, in order to provide, for example, deep semantic analysis of NL questions such as anaphora resolution, context and ambiguity detection, complex question processing, better answer ranking and answer justification, responses to unanticipated questions, response fusion or integration, and response summarization to cite just the main topics. It is also important to be able to resolve situations in which no answer is found or when the answer is incomplete or fuzzy in the data sources. This can be resolved for example via dialogue or interactive QA scenarios.

In order to address these future challenges and to guide research in this field, a number of QA roadmaps have been proposed: in 2001, Advanced Question and Answering for Intelligence, the (AQUAINT) program initiative, was created by ARDA. It was then revised in 2002 by the participants of the LREC (Language Resources and Evaluation Conference) QA workshop and further refined at the AAI (American Association of Artificial Intelligence) Spring Symposium in 2004 and by the KRAQ series 2004-2008. In these roadmaps, the research community states that QA systems should support the integration of deeper modes of language understanding as well as more elaborated reasoning schemas in order to boost the performances of current QA systems as well as the quality and the relevance of the produced answers. Advanced QA systems can be viewed as an enhancement, rather than a rival to retrieval based approaches.

In this paper, we present the main principles that we followed to identify titles, instructions, instructional compounds and arguments. From this point of view, our work is a contribution to the semantics of texts, including the recognition of some rhetorical relations, which is known to be a hard problem. This work is applied to French; we are developing a similar study for Thai. A priori, it should be possible to transpose it to a number of other languages. It is interesting to note how useful text semantics can be for question-answering.

2. How-To questions and the Structure of Procedural Texts

The main goal of the work presented here 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). The current text emerges from (Delpech et al 2007, 2008).

Answering procedural questions thus requires being 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. Analyzing a procedural text requires a dedicated discourse analysis, e.g. by means of a grammar. Such grammars (Webber 2004) are not very common yet due to the complex intertwining of lexical, syntactic, semantic and pragmatic factors they require to get a correct

analysis. Discourse grammars have basically a top-down organization, they take discourse acts as their basic units, instead of just words, they account for the structure and for the interactions between these acts and they require a relatively elaborated conceptual representation as output. Such a grammar must capture the discourse cohesion, possibly the communicative intentions, as well as the discourse organization, e.g. in terms of plans.

Procedural texts are organized sets of instructions (Adam 2001); 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 realizations, which makes identifying the structure of 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, and 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. Another feature is that procedural texts tend to minimize the distance between language and action. Plans to realize a goal are made as immediate and explicit as necessary, the objective being to reduce the inferences that the user will have to make before acting. Texts are thus oriented towards action; they therefore combine instructions with icons, images, graphics, summaries, warnings, advices, etc.

Research on procedural texts was initiated by works in psychology, cognitive ergonomics, and didactics. 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 al. 07), (Aouladomar 2005) where a preliminary structure was proposed. From a methodological point of view, our approach is based on (1) a conceptual and linguistic analysis of the notion of procedure and (2) a mainly manual corpus-based analysis, whose aim is to validate and enrich the former. Other works on this topic include (Delin et al. 1994), (Takechi et al. 2003) and (Yin, 2004).

In terms of research, analyzing procedural texts is of much interest. Indeed this is a 'genre' which is quite well restricted, but that is rich enough to allow investigations in semantics and pragmatics: rhetorical relations, temporal relations, illocutionary force, etc. It makes relatively feasible studies that would be impossible in the open language. However, results remain proper to those texts. In fact, our approach, at several levels is to try to focus on prototypical structures describing a phenomenon, and then to investigate the conditions its extension to other areas. For example, titles in procedural texts on the web are very diverse in form and contents; they nevertheless share some properties that make their tagging relatively possible. This would not be the case on open texts.

3. Basic Analysis: Recognizing Titles and Instructions

Let us now develop in more depth the different phases of our system. Preliminary steps include cleaning and labeling text objects. Then title and instructions can be recognized and tagged.

3.1. Cleaning Web texts and tagging

The inputs of our system are raw Web pages. From these pages, we need:

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

(2) to select and simplify the html tags 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 text can be considered relevant or not. It mainly uses paragraph length and proportion of close-class words criteria. We evaluated it on 100 Web pages, coming from 12 different web sites. The results were: 0,95 precision and 0,76 recall. We evaluate the recognition of titles and instruction on a hand-cleaned corpus.

The next stage is to tag the different lexical objects of the text, so that the segmentation of titles and instructions can be done properly. For that purpose, we use the Treetagger that labels all the objects (syntactic category, morphological features). We also add some semantic considerations about action verbs. Of particular interest to us is the recognition of verbs, some nouns and adjectives, modals and connectors of various kinds.

3.2 Recognizing Titles

For answering How-to questions it is obviously of much importance to recognize titles and possibly hierarchies of titles in complex texts since, in general, titles express the main goals of procedures. However, for some domains (like health) additional information should be used to index a text, in addition to titles. A first observation is that html encodings are, by far, not homogeneous. Titles are coded with the tag `<hi>` in only 20% of the cases over the 600 titles observed. In most cases the tag `` is used, possibly also `<emp>`, `<u>` and a few others (macros...). Encodings may be quite homogeneous within a given web site, but heterogeneity prevails over different sites, even in the same domain.

We identify titles in two steps. First, an algorithm processes the paragraphs of the text one by one, and gives them one of these tags: title, text or ambiguous. This first step processes easy cases. For example, an easy case for 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 given without doubt if the paragraph is subdivided into smaller units or is longer than 12 words. Ambiguous paragraphs 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 given by the first step to its surrounding paragraphs. For example, an ambiguous paragraph between two paragraphs tagged as text will be considered a title. Similarly, an ambiguous paragraph followed by a title is labeled text. We have elaborated about 7 such rules that deal with ambiguities. This second step also operates some repairs on the tags yielded by the first step. For example, any sequence marked *title* at the end of the text will be repaired as *text*. Each disambiguation/repair rule is applied sequentially and in a specific order on the list of tags.

The title hierarchy is very difficult to identify without content analysis. 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 (excluding 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 its following paragraph by a `<p>` tag. Type 2 is a title separated from its following paragraph by a `
` tag. Although we still have no means to tell the exact level 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 the 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

While working on corpora, we noted that what is usually called an instruction ranges from clearly injunctive clauses to implicit prescriptions (Talmy 2001) (this complexity is reflected in the complexity of our manual annotation tasks). Instructions are recognized (Delpech et al. 2008) on the basis of two factors: contents, around action verbs in certain forms to identify an instruction and typographic 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. We use lexical resources such as action verbs, incentive verbs, nouns and adjectives. They must have in French specific forms: imperative, infinitive, modal + infinitive, dummy pronoun 'on' + finite verb (this pattern has a semantic restriction: only action verbs are allowed), middle reflexive constructions, and gerundive forms. The frequency usage of each of these forms largely varies across domains (e.g. cooking recipes mainly use imperative while video game solutions make high usage of the dummy pronouns 'on' or even of finite forms in the first person singular). The segmenter is implemented in standard Perl. Note that English seems to have a simpler set of forms making instructions slightly more difficult to recognize.

Instructional compounds are composed of groups of instructions together with sentence fragments which are in a rhetorical relation with the instructions (advices, warnings, elaboration, etc.). They are delimited as follows: by means of typographic marks: ending of enumeration (e.g. sequences) or by 'strong' marks in long paragraphs. These marks are in general temporal (Two hours later,...), conditional expressions or goal expressions.

Finally, a grammar, based on a simple transposition of a few Minimalist Theory principles allows us to bind all the parts of the text. The grammar runs in Prolog in our prototype. The output is an XML file that reflects the text structure. Here is an extract of what we get before the grammar application, where terminal elements are tagged:

```
<p> <b> <titre>Gâteau au chocolat gourmand</titre>
</b></p>
<prerequis><p><b>
<titre> Ingrédients</titre>
</b></p>
<li> 150 g de chocolat noir </li>
<li> 75 g de beurre doux </li>
<li> 210 g de lait condensé </li> .....</prerequis>
<p>Utilisé depuis la nuit des temps, le chocolat .... si vous souhaitez l'inclure dans la
composition de votre oeuvre .</p>
<p>temps: 2 heures, assez facile. </p>
<p><b><titre> la préparation :</titre></b></p>
<li> <compinstr> <instr> 1. Tapisser un petit moule rectangulaire de papier
aluminium.</instr></compinstr> </li>
<li><compinstr> <instr> 2. A l'aide d'un couteau tranchant, concasser les amandes.
</instr></compinstr></li>
....
<compinstr> <instr> Dans une casserole à fond épais, placer le chocolat cassé en
morceaux, le beurre, le lait et la cannelle.</instr>
<instr> Chauffer doucement à feu doux pendant 3 à 4 minutes en remuant avec 1 cuillère
en bois.</instr>
<instr> Bien battre le mélange. <instr> Incorporer les amandes, les biscuits et les abricots
en remuant bien. </instr> .... </compinstr>
<compinstr> <instr> Au bout d'une heure, .... </instr> ..... </compinstr> .....</p>
```

4. Explanation structure in procedural texts

First, 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 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. All these elements are, in fact, essential in a compound for a good understanding of the procedure at stake. For example, explanations and arguments help the user understand why an instruction must be realized and what are the risks or the drawbacks if he does not do it properly. An example of an instructional compound is:

[*instructional compound* [*Goal* To clean leather armchairs,]
[*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.]]]]

Next, from our development corpus, we established a classification of the different forms explanations may take. Basically, the explanation structure is meant to help the user by making sure that he will effectively realize actions as they are specified, via e.g. advices and warnings. The main structures are facilitation and argumentation structures; they are either global (they are adjoined to goals, and have scope over the whole procedure) or local, included into instructional compounds, with a scope local to the instructional compound. These structures are summarized as follows:

- **facilitation structures**, which are rhetorical in essence (Kosseim et al 2000) (Van der Linden 1993), correspond to *How to do X ?* questions, these include two subcategories:
 - (1) user help, with: hints, evaluations and encouragements and
 - (2) controls on instructions 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 do X ?* questions. These have either:
 - (1) a positive or neutral orientation with the author's involvement (promises) or not (advices and justifications) or
 - (2) a 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 which are the most frequently encountered, besides advices.

5. Identifying arguments in procedures

5.1 General structure of arguments

Roughly, argumentation is a process that allows speakers to construct statements for or against another statement called the conclusion (Amgoud et al. 2001, 2005). These statements are called supports. The general form for arguments is : **Conclusion 'because' Support** (noted as *C because S*). Arguments may be more or less strong, they are in general associated with a certain weight. (Anscombe et al. 1981) .In the case of procedural texts, the representation is as follows.

Let G be a goal which is realized via the sequence of instructions $A_i, i \in [1, n]$, whatever their exact temporal structure is. A subset of those instructions are interpreted as arguments where the

conclusion is the instruction (*Aj*), associated with a support *Sj* that stresses the importance of *Aj* (*Carefully plug in your mothercard vertically, otherwise you will damage the connectors*).

Their general form is: *Aj because Sj* (we use here the term 'because' which is vaguer than the implication symbol used in formal argumentation, because natural language is not so radical). Supports *S* which are negatively oriented are warnings whereas those which are positively oriented are advices. Similarly to the principles of argument theory, but within the framework of action theory, if *Aj* is associated with a support of type warning *Sj* then if *Aj* is not realized correctly, the warning *Sj* is 'active' and attacks the goal *G*, i.e. it makes its realization more difficult if not impossible. Conversely, if *Sj* is an advice, it supports the goal *G*, making its full realization easier if *Aj* is executed. As can be noted, our definition includes terms which are gradual: 'more difficult', 'easier', because in practice, failing to realize an instruction properly does not necessarily mean 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 *Aj*) as well as of supports, there are many modals or classes of verbs (like risk verbs) that modulate the consequences on *G*, contrast:

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

with:

carefully plug in your mothercard vertically, otherwise you will most likely damage its connectors..

In the latter case, the goal 'mounting your own PC' is likely to fail, whereas in the former, the goal 'cleaning your leathers' will just be less successful.

5.2 Processing arguments

From the above observations, 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 texts of various domains (cooking, do it yourself, gardening, etc.). Let us focus here on warnings. 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 (Cruse 1986), (Rosner et al 1992) (sometimes we also found a few typographic marks), and then delimiting these elements. In general boundaries are either sentences or, by default, instructional compound boundaries. We have basically a unique structure composed of an 'avoid expression' combined with a proposition. The variations around the 'avoid expressions' capture the illocutionary force of the argument via several devices, here ordered 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:

- (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 nouns: *death, disease, etc.*

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, as will be seen below, some supports may be empty, because they can easily be inferred by the reader. In that case, the argument is said to be truncated.

The same strategy applies for the recognition of advices, which mainly introduce optional actions. However, they are slightly more difficult to recognize due, precisely, to their ‘optional’ character: terms are less marked, less injunctive, and therefore more difficult to identify.

Patterns are implemented in Perl and are included into the TextCoop software. We do not have space here to discuss about algorithms, but so far these are quite straightforward. From the above observations, 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. 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 89% for the conclusion recognition, and 86% for the support. Correct delimitations are 84 and 81% respectively.

Besides identifying arguments (advices, warnings) in a text and other structure, a major application of this work is the acquisition of **domain know-how knowledge**, which is probably quite basic, but which could be subject to interesting generalizations. Obviously, to make this know-how operational, it is necessary to analyze it and transform it into a formal representation that supports inference.

5. Conclusion

In this paper, we have shown how titles, instructions and arguments of various sorts can be tagged in procedural texts. This work is designed to respond to how-to questions where a well-formed portion of a text is expected as a response. Besides this level, we investigated how titles can be indexed for response retrieval (given that titles are often incomplete). A last step we are investigating is the production of the response, involving data fusion techniques as well as the development of navigation tools among and within documents when there are several candidate responses.

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.
- Anscombe, J.-Cl. Ducrot, O., Interrogation et Argumentation, in Langue française, 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.
- Cruse, A., Lexical Semantics, Cambridge Univ. Press, 1986.
- Delin, J., Hartley, A., Paris, C., Scott, D., Van der Linden, K., Expressing Procedural Relationships in Multilingual Instructions, Proceedings of the Seventh International Workshop on Natural Language Generation, pp. 61-70, Maine, USA, 1994.
- Delpéch, E., Saint-Dizier, P., A Two-Level Strategy for Parsing Procedural Texts, VSST’07, Marrakech, October 2007.
- Delpéch, E., Saint-Dizier, P., Anonymous, Investigating the Structure of Procedural Texts for Answering How-to Questions, LREC 2008, Marrakech.
- Davidson, D., Actions, Reasons, and Causes, Journal of Philosophy, 60, 1963

- Kosseim, L., Lapalme, G., Choosing Rhetorical Structures to Plan Instructional Texts, Computational Intelligence, Blackwell, Boston, 2000.
- Rosner, D., Stede, M., Customizing RST for the Automatic Production of Technical Manuals, in R. Dale, E. Hovy, 2002.
- D. Rosner and O. Stock eds., Aspects of Automated Natural Language Generation, Lecture Notes in Artificial Intelligence, pp. 199-214, Springer-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.
- Van der 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.