Semi-structured Documents Reengineering

Carmen López-Rincón,
Jaime Sarabia Álvarez-Ude

Under different names (Document Analysis and Understanding, Document Reengineering... [Brugger & al. 1997], [Klein & Fankhauser 1997], [Litman 1996], the analysis of informational text structure is emerging as an important technology in the field of linguistic engineering. In the following pages we treat second-order bibliographic document analysis as a paradigmatic example of semi-structured documents, introducing a set of well-tested techniques for their efficient parsing and reengineering, rooted on the logic-programming paradigm. Unlike other approaches to document information analysis, we try to deal with issues related not only to theoretical coherence but also to real applicability. We first define the abstract class of documents and then we discuss their parsing and some implementation issues.

1 SEMI-STRUCTURED DOCUMENTS

1.1 Text structure

Our main case study is second-order documents, that is, documents containing information about other documents. Typical cases are library catalogue cards or tables of content. Although these are our primary motivation, we think the ideas here presented have a much wider application field. The actual document processing has led us to an abstract view of document structure which is suitable to other classes of semi-structured documents We have got a fair evidence supporting this hypothesis, although inconclusive in relation to the boundaries of the tractable document class.

We will first use very loosely the notion of documentary or textual structure, trying to clarify the concept by some examples. Later we will work a more formal definition, derived from the analytical tools we use in its parsing.

Let us consider Figure1, a bibliographic record. As any other document is a linguistic object and one or several linguistic structures can be found in it. In addition it has, alongside its linguistic structure, a textual or documentary organisation. In other cases a document is perhaps organised in paragraphs, headings, footnotes, etc. In this case a list of keywords (*Título, Entidad* …) introduces specific areas. This area structure is the text structure of the document.

Characteristic of this and other semi-structured documents is that linguistic structure (as seen from the grammar point of view) does not explain its meaning. It does not even explain how it is possible to convey information with such a text. So in Figure 1, for instance, the traditional parsing can discover at most a

Título: Anales de documentacion : Revista de Biblioteconomia y Documentacion
Entidad: Universidad de Murcia. Escuela Universitaria de Biblioteconomia y Documentacion
Publicacion: Murcia : Universidad. Escuela Universitaria de Biblioteconomia y Documentacion, 1998-
Peridiocidad: Anual
Materia(s): Biblioteconomia Publicaciones periodicas; Documentacion Publicaciones periodicas
CDU: 02(05); 002(05)
Número de control: X533246783

Figure 1

list of unconnected name phrases. But note that

- The document as a whole has a meaning of its own
- Text meaning is compositionally organised
- The meaning of its areas is determined by its role in the document.

Now the question naturally arises, how can these texts be parsed? Let us delay a concrete answer to this question to the second part of this paper and look a bit longer to semi-structured documents specific features.

Text structuring -formatting, text layout- serves several purposes, ranging from readability -layout of headings, for instance- to meaning determination -title, authorship, and pages in Figure 2-. Typically, the more repetitive is a class of documents, the more meaning is conveyed by structural -format, layout- properties and more essential becomes text structure interpretation for text understanding: formatting expresses in a compact way recurrent meaning substructures. Therefore textual organisation, by itself, can convey information relevant to document meaning. Semi-structured documents are texts in which this possibility is explicitly used.

Both structure mark-up and level of organisation greatly varies among document classes. So both Figures 1 and 2 show quite strong structure mark-up, though through different devices. In Figure 11 words in italics mark document structure

[1] Taken from the library catalogue of the Universidad Complutense de Madrid.

while in Figure 2 structure is displayed through a mixture of words and language independent marks: graphical layout, punctuation, etc. We will call layout mark-up this type of structure marking. Layout mark-up is (almost) meaningless[2] when considered from the natural language point of view, but contributes nonetheless to text meaning determination when used as part of an -often implicit- text structuring code.

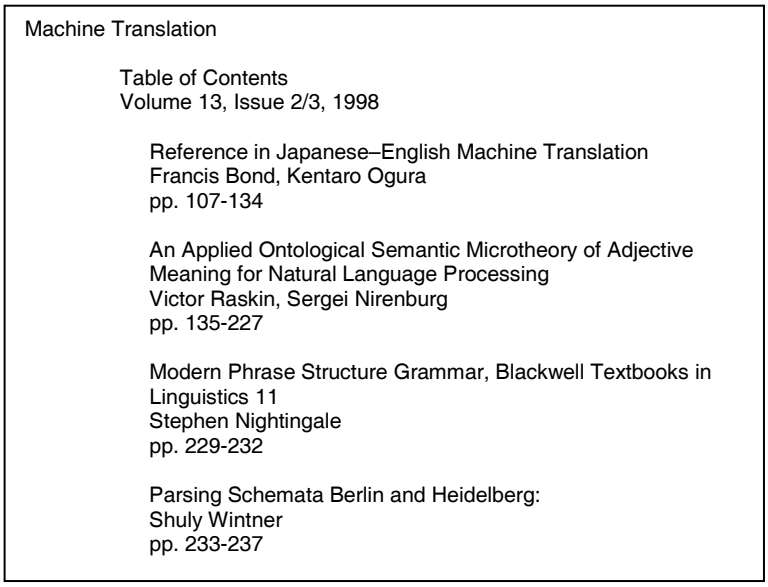

Figure 2

On the other hand (1), (3) and (4) below show a minimal but important layout mark-up: as in any text, you will find meaningful segments -words, for instance- intermixed with marks whose function is to define how the text is build. Consider (1):

(1) This is not a pipe but a sentence.

Its linguistic structure could be represented according to your preferred linguistic theory, typically as a tree with root in 'sentence' and nodes like noun or verb phrase, etc. Terminals -elementary meaningful segments- in this structure are words (*This, is, not, a, pipe*). Note however the layout mark-up intermixed with words: upper case characters, dot and spaces. These marks are not accounted for in standard linguistic parsing but they fulfil a crucial function: they mark the text beginning or end and define the elementary components the text is made of.

[2] Which meaning has an indentation?

Again, textual mark-up partially determines text meaning, although the marks themselves are not meaningful: colon and comma are not syntactic but textual objects.

Therefore layout mark-up is a very convenient device, as shown by (2)

(2) Thisisnotapipebutasentence.

But not only that: layout mark-up is necessary condition of meaning determination:

(3) lo hice con Pilar / lo hice compilar[3]

(4) es con pasión / es compasion[4]

(3) and (4) are homophonic but not ambiguous, when written.

Let us return to semi-structured documents. For instance, the bibliographic card in Figure 3. In these cases layout mark-up plays a fundamental role in determining the information displayed in the text: upper case characters on line 2 mark the function of the name; indentations of lines 3, 6 7, 8 represent end of information blocks. And these blocks contribute to meaning determination of words: the first occurrence of *III* in line 6 (*III + 457 p. ; 28 cm*) stands for Latin numbered pages, while the second (*III. Título*) has a wholly different, meta-linguistic, meaning: the ordinal of a secondary access point. Note also that English and Spanish are used: text structure validates this mixture, otherwise unacceptable.

```
1     61 ABB-1
2       ABBAS, Abul K.
3       Cellular and molecular immunology / Abul K. Abbas, Andrew
4     H. Lichtman, Jordan S. Pober .-- 2nd ed. -- Philadelphia :
5     W.B.Saunders Company, cop. 1994
6       III, 457 p. ; 28 cm
7       Incluye bibliografía e índice ISBN 0-7216-5505-X
8       1.Inmunología Celular. 2.Inmunología molecular. I. Licht-
9     man, Andrew H. II. Pober, Jordan. III. Título.
1     616.07'9
0   R. 438
```

Figure 3

In the same vein, the structure mark-up of the field

Publicacion: <text1> : <text2> <date>

[3] I did it together with Pilar / I ordered it to be compiled
[4] It is with passion / it is pity

in Figure 1 determines the meaning of the text segment as a whole and that of their constituents: for instance, <text1> should be the name of a place, <text2> the publisher's name and <date> the earliest date for copies to be found in the library.

In general, semi-structured texts have well defined, foreseeable structures. Here layout mark-up can improve readability, as usual, but above all it makes possible efficient ways of expression. Recurrent text parts are coded into a text structure that conveys the same information as a wholly explicit text but with a very restrained set of expressive means.

Layout mark-up is naturally intermixed with words, although its role is very different from theirs. In a sense, text formatting defines data structures while the text itself is the data or contains the data.

This distinction is by no means a sharp one: in some cases there are text segments that play a dual role, as bearers of information (data) and also as text structure markers. Compare for instance the fragments of two bibliographical cards in Figure 4: the first presents an explicit textual mark-up dissociated from the data

. - <text1> . - <text2> : <text3> , <text4> <end_of_line> <text5> p. : <text6> …

Here it is possible to interpret the function of the text segments almost only from the layout mark-up: given the 'beginning / end of field' mark ('. - ') and its logical position in the card it is possible to foretell that <text2> is the name of a place while <text3> is a publishing house[5].

[more text] . – second printing . – London : Pall Mall Press , 1.968 350 p. : fot. neg. ; 22 cm . [more text]
[more text] (Coll. de Textes pour servir à l'étude et à l'enseig. de l'Hist.) Paris. Picard et Fils. 1897. XXXVI–290 pp [more text]

Figure 4

[5] According to the ISBD format

In the second case *Paris* is used both as a layout marker -together with indentation and end of line- and as data. Some natural language knowledge is then needed in order to parse textual structure [López Rincón 1995].

So in a semi-structured document a document type specific structure -laid over natural language segments- largely characterises the document information as well as that of its parts.

Besides, global text coherence (the possibility of its conveying a meaning or -what amounts to the same- being an instance in a document class) rely on the document actually having and structure. So in Figure 5 the fault structure of Part A makes it highly ambiguous and so not immediately understandable, contrary to Part B.

`Edited by Elizabeth A. Liv-ingstone Kalamazoo Michigan, Cistercian Publications, 1985` Part A	`Edited by Elizabeth A. Liv-ingstone .— Kalamazoo, Michi-gan : Cistercian Publica-tions, 1985.--` Part B

Figure 5

2 SEMI-STRUCTURED DOCUMENTS PARSING

We turn now to the actual tools we are using to parse semi-structured documents ([López Rincón 1995] - [López Rincón & Sarabia 1999], [Sarabia 1992] - [Sarabia & López Rincón 1999], [Verba Logica 1996]). Our approach to the problem has been to devise a very high level language for semi-structured document description: LENDEX. A LENDEX description of a document class represents its (text-)grammar. It becomes a parser for the document class and yields as output a parse tree for text structure. Now let us briefly review the methods we use to perform document description and parsing.

2.1 LENDEX grammars

A LENDEX grammar begins with a production listing the areas in a document collection in the form:

```
Class ==> Area1,....,AreaN.
```

where `Class` is an atom representing the document class name.

This production is followed by the definition of the areas -`Area1,....,AreaN`- from which the documents in the class are built up.

Now LENDEX classifies areas according to two criteria:

a) weakly /strongly defined areas: a weakly defined area (W-area) is a text segment marked just by the kind of ending it has. It is strongly defined (S-area) if it has an internal area structure of its own.

b) R-areas / RR-areas: a recognition only area (R-area) spans over a text segment relevant only for recognition purposes: typically, it marks structure but has no other informational content. On the other side, a recognition and retrieval area (RR-area) spans over a segment which not only (possibly) defines text structure but contain information relevant for the intended document analysis.

Figure 6

An example: the text structure of Figure 6 is represented by the areas-tree in Figure 8. Figure 7 displays a LENDEX grammar for this kind of document.

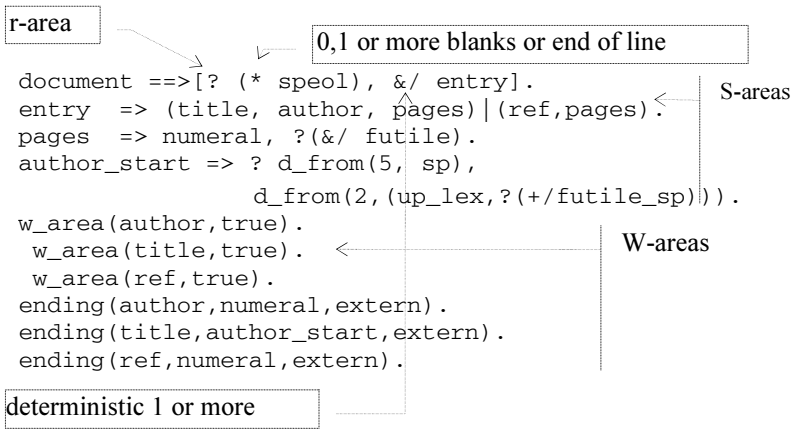

Figure 7

Its main production says that documents contain one or more entries, after possibly some blanks and end of line marks (to be recognised but not saved at final analysis). Then an entry is either a title followed by an author and a page, or a reference (*Books Reviews*,...) followed by a page. Author is an area whose ending mark is a recognition-only sequence of at least 5 blanks followed by at least two upper case lexemes, etc

The areas named `author`, `title` and `ref` in the grammar are W-areas: they are characterised just by its ending. On the other side, `entry`, `pages` and `author_start` are S-areas: they are defined through parsing of their internal structure

Besides, note that some text segments in the original text have disappeared in this areas tree: so for instance the sequence `* speol`[6] between title and author name. `* speol` is an R-area. This fact is marked by the prefix operator `?`.

These ideas are mirrored in LENDEX by means of different constructions:

1. W-areas definition essentially involves stating the class of areas which acts as ending mark and its type (internal or external to the area) and optionally defining a condition that every significant lexeme in the area must satisfy.

   The end of area definition is in fact a complete area and contains the same expressions as the body of S-area, as stated below.

2. S-areas are represented in terms of productions of the form:

```
Area_Name => Area_def.
```

where `Area_def` is the body of the production and may contain several kinds of constructs such as, among others:

a) sub-categorisations of the lexeme concept (roughly, terminal), discriminating between lexemes in upper case character or beginning with such a character, numerals, etc.

b) sequences of terminal lexemes,

c) iterative operators similar to these used in regular languages, for instance 'optional' or 'one or more' and other numerical quantifiers such as 'at least n ', 'at most k ', etc,

d) standard Definite Clause Grammar notation and PROLOG syntax, the programming language underlying LENDEX.

[6] Meaning the same as {<space> | <end of line> } in EBNF

3. The distinction between R-areas and RR-areas is established in several ways. To be 'recognition only' can be stated as a global property or just affecting an occurrence of an area. Besides, the text segment covered by an R-area can be wholly lost in the final areas tree, or subsumed in its parent area.
4. Additional constraints can be stated which are to be met by areas as a whole. According to their being strong or weak, falling to meet a condition implies a failure for the text segment to be characterised as an area of certain kind, or just a warning on the correctness of this attribution.

2.2 The Parser Engine

The Parser Engine is a module performing a double function: it codes LENDEX descriptions as parsers and applies them to the relevant documents. Therefore two subsystems can also be distinguished: The first acts as a compiler, translating LENDEX expressions into standard PROLOG code. The second sub-system is an enhanced grammar evaluation system. It drives the actual parsing of documents with two main objectives: first, to support a robust parsing strategy. Second, to generate an areas-tree to each document in the class.

The first goal answers a major problem in the standard search engine of DCGs: its inability to overcome -at least partially- a parsing failure. The characteristic unstability of document collections makes it necessary a more robust parsing strategy. The Parser Engine is able to cope with three kinds of problems:

p1) Undetermined text segments: these are segments not parsable by any appropriate category.

p2) Missing categories: mandatory categories for which no appropriate text segment is found.

p3) Inaccurate categories: categories whose relevant extension and / or informational content do not satisfy some constraint imposed on it.

So the Parser Engine exercises a degree of self-control, reporting on problems encountered during parsing and greatly simplifying the output integrity evaluation.

2.3 Text structure: an areas-tree

The output of the Parser Engine on a LENDEX grammar and a document is a finitely generated tree whose nodes are areas. Informally an area is (a transformation of) a labeled segment of text or a labeled areas-tree. It is this areas-tree which displays the textual structure of the document.

In order to understand our view of this structure, three concepts are needed: 'actual extension', 'relevant extension' and 'information content' of the document.

A LENDEX description or grammar is associated first with a tree in the usual sense of a 'parse tree': each category in the grammar produces a node which either dominates an original text segment (a leaf) or is the root of a tree in the same sense. We call 'actual extension' of a category either the text segment itself -in the case of a leaf- or the concatenation of the actual extensions of the categories dominated by it, otherwise.

```
entry   title   Pictures and Truth
        author  KORNEL SOLT
        pages   154
entry   title   The Art Object
        author  BARBARA E. SAVEDOFF
        pages   160
entry   title   Intentional Semantics and the Logic
                of Fiction
        author  DALE JACQUETTE`
        pages   168
entry   ref     Books Reviews
        pages   177
entry   ref     Books Received
        pages   193
```

Figure 8

Second, actual extensions are mapped in relevant extensions by: a) trimming the actual extension tree leaves and b) substituting the empty text for the leaves dominated by a R-area. The relevant extension for a category is now defined similarly to the actual extension above.

Finally, the relevant extension tree is mapped in the areas tree, representing the document 'information content' (IC) and its textual structure. Its main features:

Only W- and S-areas characterised as RR-areas appear in this tree. W-areas are presented as pre-terminal nodes, while S-areas are recursive trees.

The IC of constructs such as DCG defined categories, marked lexemes, sequences of lexemes coincide with their relevant extension. The IC of an iterated category is a list containing the ICs of the category occurrences.

Some meta-linguistic marks -‘undetermined', ‘ignotus’ and ‘incorrect’- can be produced by the grammar evaluation system in order to qualify incidents in the parsing process, such as those alluded in p1), p2) and p3) above.

3 IMPLEMENTATION ISSUES AND APPLICATIONS

We have used mainly logic programming for implementing the above-presented ideas. It offers us tools such as intensive meta-programming, program transformation, etc. which are hardly available in other programming languages. Such facilities allow for some nice LENDEX features: for instance, the different output trees alluded above are generated automatically without explicit mention to them in the grammar.

The system has been mainly used in Automatic Retrospective Conversion of library catalogues, including printed catalogues, catalogues on typewritten paper cards or on magnetic media of many sorts, etc. Classified advertisements, dictionaries and other sorts of documents have served also as test cases for the system general applicability.

The most recent application of these techniques has been Web pages analysis, intepreting HTML as layout mark-up, that is to say, extracting information from mark-up mainly thought as a visual device.

Let us be explicit about two advantages of our approach: very fast development of a document structure model and easy program maintenance. Hence, the cost of writing a parser for document class is acceptable even if the collection is very small. Of course, as the structure of documents gets more complicated and / or the collection gets bigger, the benefits of using a tool like LENDEX increase.

3.1 Future Directions

One immediate aim is to extend LENDEX applicability in other fields of Information Extraction, trying to use structural information together with tools of a more 'linguistic' flavour, some of them already integrated in the overall LENDEX architecture. So for instance, a variety of noun phrases, such as proper nouns, dates, etc. can be retrieved through LENDEX grammars in a very efficient way, and obviously melted with structural information.

A LENDEX description of a document class is akin to some SGML constructs, particularly to XML DTDs. Although it originated independent and previously to its definition, translation to and from these DTDs to LENDEX grammars seems to be feasible.

We work also in a further step in automated semi-structured document processing: some experiments in grammar learning have suggested a possibility of automatic deriving LENDEX grammars from a document sample.

REFERENCES

Brugger - Zramdini - Ingold.- *Modeling Documents for Structure Recognition Using Generalized N-Grams*. 4th International Conference Document Analysis and Recognition. (ICDAR'97)

Linden.- *Structured Document Transformations*. Ph. Thesis Series of Publications A, Report A-1997-2.Helsinki, 1997.

Klein - Fankhauser.- *Error Tolerant Document Structure Analysis*. International Journal on Digital Libraries. vol I, 4. 1997

Kuikka. *Processing of structured documents using a syntax-directed approach*. Kuopio University Publications C. Natural and Environmental Sciences 53. 1996. 76 p.

Dengel - Kieninger..- *SALT : Document Structure Analysis based on Layout and textual features in mixed-mode documents*. http://www.dfki.uni-kl.de/da/id_pro.html

Litman .- *Cue Phrase Classification Using Machine Learning*. Journal of Artificial Intelligence Research 5 (1996) 53-94

López Rincón.- *Diseño e implementación de un sistema de análisis documental automático*. Madrid: Tesis doctoral UCM, 1995

López Rincón.- *LENDEX: un lenguaje para el análisis documental automático.* En *XII Congreso de lenguajes naturales y lenguajes formales*. Barcelona, PPU, 1996.

López Rincón.- *Un agente inteligente de extracción de información basado en programación lógica*. Actas 2ª JADOC. Granada, Spain. Noviembre 1999, pp. 373-389

López Rincón - Sarabia.- *Un sistema de conversión retrospectiva automática basado en el análisis lógico de documentos semi-estructurados.* En: Colima, Mexico. Noviembre 1999

F.Parmentier and A.Belaïd, *Bibliography References Validation Using Emergent Architecture*, ICDAR'95, Vol. II, p. 532-535, Montréal, Québec, August 14-16, 1995.

Sarabia.- *Técnicas de metaprogramación en gramáticas lógicas*. En VII Congreso de lenguajes naturales y lenguajes formales. Barcelona, PPU, 1992.

Sarabia.- *Análisis textual automático: proyecto BiblioTECA*. En XII Congreso de lenguajes naturales y lenguajes formales. Barcelona, PPU, 1996.

Sarabia – López Rincón.- *Automatic Information Analysis for Second Order Documents Retrospective Conversion***.** In: Proceedings of the Symposium on Industrial Applications of PROLOG, Sept. 1997. Kobe University. Kobe, 1997 pp. 20-28.

Sarabia – López Rincón.- *Extracción y procesamiento automáticos de información. Una aplicación en conversión retrospectiva*. In Actas 2ª JADOC. Granada, Spain. Noviembre 1999, pp. 345 - 357

Smith - López.- *Information Extraction for Semiestructured Documents*. In Workshop on Management of semiestructured data. Tucson, Arizona, Mayo 1997.

Verba Logica.- *Biblioteca project*. http:// www.ucm.es/info/VerbaLogica /biblio.htm

Carmen López-Rincón is Associated Professor at the Logic Department of the Complutense Universidad of Madrid (Spain), Ciudad Universitaria, s/n. 28040 Madrid. She can be reached at clrincon@eucmax.sim.ucm.es

Jaime Sarabia Álvarez-Ude is Profesor Titular at the same department. He can be reached at sarabia@eucmax.sim.ucm.es.

The work was done under partial support of contracts from the European Union (BiblioTECA project Lib-2023) and the DGCYT and the CYCIT, research funding agencies of the Spanish government.