Information extraction for semi-structured documents

Dan Smith[†] and Mauricio Lopez

GIE Dyade, INRIA Rhône-Alpes,
655 Avenue de l'Europe, 38330 Montbonnot.

Dan.Smith@inrialpes.fr, M.Lopez@dyade.fr Tél : (33) 04-76-61-52-39 Fax : (33) 04-76-61-52-52

1. Introduction

The number of unstructured or semi-structured documents produced in all types of organizations continues to increase rapidly. Cost-effective ways of finding the relevant ones and extracting useful information from them are increasingly important to a large number of enterprises for operational and decision-support applications. The approach discussed in this paper constitutes a suitable basis for building an effective solution to extracting information from semi-structured documents for two principal reasons. First, it provides an extensible architecture basis for: extracting structured information from semi-structured documents; providing fast and accurate selective access to this information; performing selective dissemination of relevant documents depending on filtering criteria. Second, it is simple in terms of: the complexity of the algorithms used for structure recognition and document filtering; the number and size of data structures required to perform the three functions mentioned above; the amount and complexity of the metadata required to handle a given collection of documents. The work described here is part of the Dyade[1] Médiation project, which aims to provide integrated software components for accessing heterogeneous data sources in Internet/Intranet environments.

The primary focus of the work described here is the need to extract and disseminate information from heterogeneous sources of semi-structured documents in a particular application domain. The motivational example is that of Calls for Tender - Appels d'Offre (AO) for public works in France. These documents contain details of the nature of the work: commissioning authority, location of the work, object of the call, etc. Their content is organized in a similar way, but a given piece of information is not reliably present in every document and documents containing similar (or the same) information may be structured differently; they are typically small. Many classes of document with similar characteristics exist in virtually every organisation which we know.

These features have allowed us to adopt an effective structured approach to information extraction, implemented in InfoExtractor, a tool which recognizes the implicit structure and identifies pertinent information, *concepts*, contained in semi-structured documents. When InfoExtractor analyses a document, the result is a structured version of the concepts it contains. In our example AO document[2] (Figure 1), the concept DatePublished is coupled to the *concept instance* "06/06/96"; Department to "AISNE"; Object to "construction of 18 rented apartments..."; etc.

The approach followed is based on a document abstract model in which the document is partitioned into regions, which we term *evaluation contexts*, where only certain concepts are searched for. Concept identification is guided by *concept definition frames* which contain the information needed to find an instance of a given concept within an evaluation context. A concept instance can itself be considered as an evaluation context where other concepts can be identified, giving a hierarchical organization of concepts. The corresponding partitioning of the document into a hierarchy of contexts, each corresponding to a

---

[†] On sabbatical from School of Information Systems, University of East Anglia, Norwich NR4 7TJ, UK

[1] Dyade is a consortium between Bull and INRIA aimed at technology transfer (http://www.dyade.fr).

[2] All the examples have been translated from the original French.

limited structural and semantic domain, allows us to avoid the use of complex natural language understanding techniques during the concept identification process.

Three problems must be addressed in the design of an information extraction tool: homonyms, synonyms, and that some words are only useful discriminators in certain phrases or contexts. For example in the AO domain concepts are typically represented by section headings (Object of the offer, Number and composition of the lots, Project manager, ...). Some of these have relatively fixed content (e.g. Department) but the content of others can be very variable (e.g. Lots). We have explicitly addressed these issues in the design of InfoExtractor

The advantages of our approach are that: it provides a basis for easily implementing the information extraction requirements of a new domain; it analyses documents for information dissemination applications which couple good precision with high recall; it can be used to supply structured data for data warehouses, information retrieval systems, etc.

In the remainder of the paper we describe our document abstract model and the techniques used for concept identification and document structuring. We conclude with a summary of our contribution and an outline of future work.

2. Related work

Information mediation research issues are being addressed in several projects, such as COIN [GOH97], TSIMMIS [CHAW94], Information Manifold [LEVY96]. The focus of the Dyade Médiation work is on the development of components to permit the rapid development of applications which combine and extract information from heterogeneous sources and integrate with current systems to maximise the value of existing investments. This work is based on the DISCO [TOMA96] architecture to provide uniform access to distributed heterogeneous structured and semi-structured data sources.

The treatment of semi-structured data sources is the subject of substantial research attention, largely because much information available on the Web is semi-structured. A recent survey [ABIT97] notes that there is no theory or precise definition of semi-structured data. A document may contain its own metadata, but the common case is for the logical structure to be implicitly defined by a combination of physical structure (e.g. HTML tags, line and paragraph boundaries) and content indicators (e.g. words in section headings). The extraction of useful content has been tackled from two major perspectives in recent research efforts: from a natural language processing perspective (e.g. [COWI96], [LESJ96], [RILO94], [SODE95]) and through extended database query languages (e.g. [AQMW97], [BUNE96], [MEND96]). Our approach, which shares elements with both these perspectives, is distinguished by the explicit use of an abstract document model to structure and restrict the amount and complexity of processing required to extract information from semi-structured documents.

Our focus is on information extraction and dissemination, so we do not suppose the availability of a reference corpus; hence our approach is based on identifying a set of predefined concepts that are present in the documents. This is closely related to work on document classification, which characterises a document as relevant or not according to the terms it contains compared with a collection of documents (e.g. [LEWI96], [SALT94], [SALT93]).

Information extraction projects typically assume that there are many documents that will meet a user's requirements and that they are better met by returning a small number of relevant documents (high precision) [RILO94], [HARM95], [GRIS95] at the expense of better recall. In the application domains we are concerned with, good recall is essential, even at the expense of lower precision.

A critical issue for all intelligent information retrieval and natural language processing systems is how much manual work is required to furnish the initial concepts and train the system on an appropriate set of

documents. The time required to manually construct a dictionary of patterns and rules for the linguistic analysis of a domain is substantial. Attempts to reduce the effort required rely on a tagged training corpus [RILO94], [SODE95] or a pre-classified document collection [RILO95]. These approaches, and several other systems used in the MUC evaluations, rely on the linguistic analysis of parts of the text identified by trigger words. One of our major design aims is to reduce the software complexity of the analysis tools and to provide a simple set of tasks for the knowledge engineer configuring the system for a new document class.

3. Document abstract structure

From the perspective of Médiation, a document can be characterised as consisting of one or more evaluation contexts, which are identifiable physical sections (e.g. header, body, summary), which vary according to the document class. Thus, a simple class of documents might only have a header and body, whereas a more complex class may have a header, several body sections dealing with different aspects of the subject, and a summary. An evaluation context contains text or other multimedia content which describes a number of concepts of interest, as well as other irrelevant material which must be ignored.

Concepts are hierarchical. The top level in the hierarchy is formed of concepts obtained directly from the document. Lower levels are formed of concepts whose evaluation context is the instance of a higher level concept.

The motivation for taking this view of documents and concepts is that linguistically similar but semantically different constructs may occur in different places in a document. By exploiting the structure of the document to restrict the scope of evaluation of a particular concept to a single context we simplify the knowledge engineering effort required for a class of documents.

The naming and scope of evaluation contexts is part of the analysis required for the application domain. Figure 1 shows an example document, with two evaluation contexts: 'header', and 'body'.

The starting point for an evaluation context is the start of the last concept instance that was identified in the previous evaluation context (arrowed in Figure 1). We have implemented this strategy because it is generally more difficult to specify and recognise end points than start points in documents. However, if there is a well defined end point for a concept this can be given and the evaluation will continue from there.

It is probable that fragments of some concepts will be located in different evaluation contexts. To accommodate this we allow a concept to appear in multiple contexts.

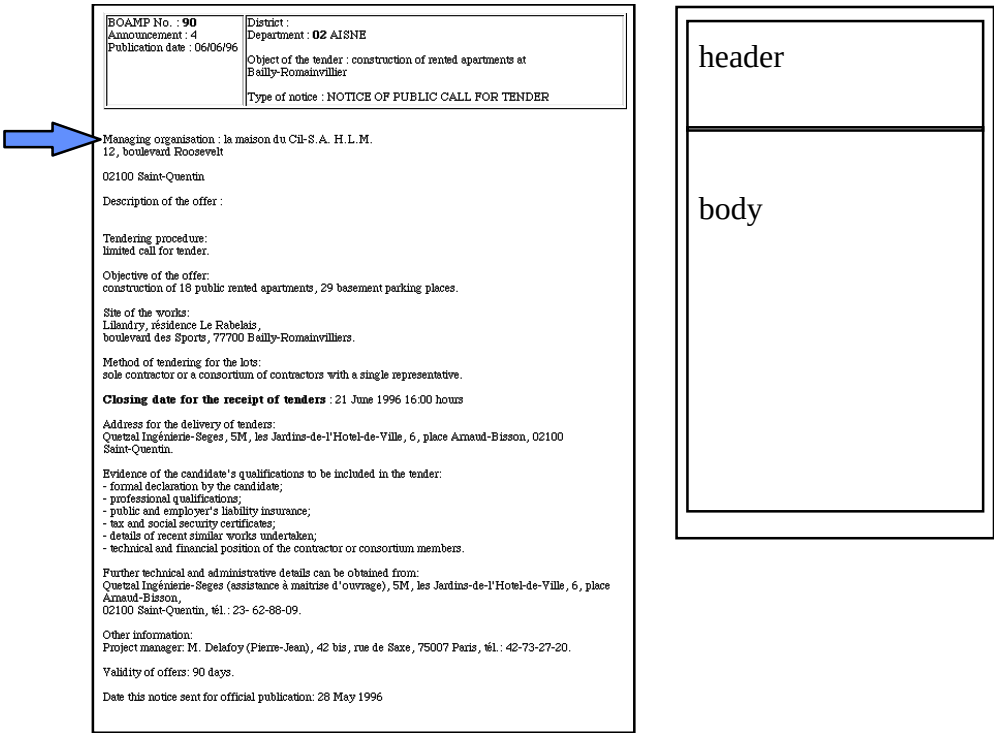

Figure 1. *Relationship of evaluation contexts to document.*

4. Information identification and structuring

4.1. Overview

The basic operation of InfoExtractor is, for each document, to take an evaluation context (e.g. Header), retrieve the definition frames for the concepts that may be present (e.g. Department), sort them into *evaluation groups* and evaluate each group in turn. The concept instances in an evaluation group are non-overlapping, so that a portion of a document can only be represented by one concept in the group. However, concept instances in different evaluation groups in the same evaluation context can overlap, since they all start from a common point. This allows the information in a document to be presented in terms of a network of overlapping concepts, which is essential for more complex domains that we have studied (e.g. medical records). After a document has been searched for all the concepts applicable to a context, the search of the next context is started from the beginning of the last concept instance that was found.

When the analysis is complete the instansiated concept frames constitute a structured version of the relevant information contained in the original document which can be used as input to match against stored interest profiles. Concept instance frames can also be stored and indexed, for use in answering future queries. For example, the interest profile of an enterprise interested in calls for tender in the Lot or Lot et Garonne départements would contain a condition like "Departement *contains* 'Lot'" which provides a better precision than the condition "*contains* 'Lot'" that could be applied to an unstructured document[3]. A high level of recall can be obtained by an adequate combination of conditions of this kind.

[3] Note that AO documents frequently contain lists of the lots into which the tender is divided, "Lot 1: reinforced concrete...". This poses obvious problems.

The approach adopted for querying the concept instances uses the same principles. The DISCO query language has an OQL-like syntax with a CONTAINS predicate to allow approximate keyword matching. This allows queries to be specified with specific concepts of interest and enables approximate matching of the terms in the concept instance. We achieve this through a binary vector model coupled to a preliminary (optional) transformation of terms via a thesaurus.

4.2. Concept recognition

The information required by InfoExtractor to identify a concept is contained in a concept definition frame, an example of which is shown in Figure 2.

concept_name	**ProjManOrg**	*The name of the concept.*
eval_context	**Body**	*The evaluation context for the concept.*
ident_pattern	**/^\s*[A-Z]/ && /:/** [4]	*A regular expression that defines the possible start of the concept instance.*
ident_keys	**responsible (0.25), organisation (0.33), works (0.5), managing (0.5)**	*The keywords and weights used to identify the presence of a concept instance.*
end_pattern	**<null>**	*A regular expression that defines the end of the concept instance.*
eval_order	**1**	*Evaluation group for the concept.*
eval_status	**Optional**	*Whether or not the concept is necessarily present.*
domain	**AO**	*List of document classes and concept names that indicates the scope of applicability of the concept.*

Figure 2***. Concept definition frame.***

InfoExtractor supports two types of expression for concept recognition: simple and complex.

A simple recognition expression is a regular expression which exactly specifies the trigger tokens and keys for a concept and for which the concept body is found entirely on the same line of the document. Dates are usually described in this way. The target of the recognition - the concept instance - is specified in the regular expression. When InfoExtractor matches this the target is assigned to the body slot of a *concept instance frame* and the search continues for the next concept. Simple expressions do not have a matching score, as they are exact matches.

A complex recognition expression specifies a pattern which indicates that a concept may be present; for instance in the example the first letter on a line is a capital and the line contains a colon. The pattern may be a trigger for several concepts. If a pattern is found, the line is compared with all the candidate concepts which have that pattern as a trigger and the line is searched for keywords that identify candidate concepts. Each keyword has an associated weight in the range 0-1. It can be manually altered, but the default weight is the reciprocal of the number of occurrences of the word in the current evaluation group. This is calculated (or given, if the calculated value is inappropriate) during the construction of the concept definition frame by the knowledge engineer. The weights of matching keywords for each concept are summed and the highest scoring concept is selected.

After a concept is selected, the text used to identify it is discarded and subsequent text is added to the concept body until the end of the concept is detected, either by recognising an explicit end-of-concept pattern, or by matching the start of another concept in the evaluation group. In the example, on lines indicating the start of a concept, the concept body starts immediately after the colon (:), as this is the last item specified in the concept identification pattern. If no concept matches sufficiently well the current concept is retained.

---

[4] This matches a line where the first non-space character is an upper case letter and which contains a colon.

Extraneous material (e.g. irrelevant sections) can be recognised and ignored by specifying a concept with the special name "null". Following the recognition of the null concept any content is ignored until another concept is recognised.

The dialogue for testing and developing the concept identification and extraction process for the example document is shown in Figure 3; it is intended for testing and developing the concept specifications and is not an end-user interface. In the current prototype the instantiated concept instances can either be taken by another application for storage and further processing, or they can be transformed into a term list and matched to user interest profiles.

```
Context: Header
Group:   1 (starting line 1)
Concept list:
  District
  Department
  NoticeType
  DatePublished
```

```
Matches found:
 Line Concept            Match% Matching text
   14 DatePublished          -  06/06/96
   17 Dept                   -  AISNE
   21 NoticeType             -  NOTICE OF PUBLIC CALL FOR TENDER
Evaluation of Header (group 1) finished on line 23
```

```
Context:  Body
Group:    1 (starting line 24)
Concept list:
  SelCrit
  DateSent
  StartDate
  CandQual
  Chars
  Duration
  ProjMan
  Lots
  Object
  ValPeriod
  CloseDate
```

```
Matches found:
 Line Concept            Match% Matching text
   27 ProjMan            50.00% Managing organisation : la maison du Cil-S.A. H.L.M
   38 Object             73.17% Objective of the offer:
   42 WorkSite           68.20% Site of the works:
   47 CloseDate          75.00% Closing date for the receipt of tenders : 21 June 1996 16:00 h
   55 CandQual           33.33% Evidence of the candidate's qualifications
                                to be included in the tender::
   73 ValPeriod         100.00% Validity of offers: 90 days
Evaluation of Body (evaluation group 1) finished on line 77
```

Figure 3. *Example output from InfoExtractor.*

5. Conclusion and future work

We have achieved our aim of a design that can be readily adapted to new domains by: only operating on a portion of the document in each major iteration, allowing the evaluation order of individual concepts or groups of concepts to be specified, allowing the initial pattern matching to be decoupled from the matching of keywords to determine which concept has been discovered.

An important direction of future work will be to extend the model and functionality of InfoExtractor to deal with multimedia documents.

The principal advantage of extracting information by successive refinements is that it keeps the main selection functions clean and allows the messy details - which are often important recognition cues - to be introduced where they are most likely to be relevant. In the case of unstructured text, high precision

requires substantial sophisticated analysis of the sentences that may be relevant. The hierarchical approach we follow addresses the problems of word-based approaches by successively decomposing the document using relatively simple criteria, which restrict the scope of meaning that a text fragment may have, making subsequent retrieval and dissemination more precise.

We anticipate that these features will allow us to build flexible cost-effective tools. In particular, it should be possible to constitute a complete range of versions providing incremental functionality and performance with respect to the size of the document base, the complexity of the document contents considered and the complexity of the filtering and selection criteria.

References

[ABIT97] S. Abiteboul, Querying Semi-Structured Data, Proc. ICDT '97, to appear.

[AQMW97] S. Abiteboul, D. Quass, J. McHugh, J. Widom and J.L. Weiner, The Lorel Query Language for Semistructured Data, *J. Digital Libraries*, to appear

[BUNE96] P. Buneman, S. Davidson and G. Hillebrand, A Query Language and Optimization Techniques for Unstructured Data, *Proc. ACM SIGMOD Int. Conf. on Management of Data*, 505-516, 1996

[CHAW94] S. Chawathe, H. Garcia-Molina J. Hammer, K. Ireland, Y. Papakonstantinou , J. Ullman and J. Widom, The TSIMMIS Project: Integration of Heterogeneous Information Sources, *Proc. IPSJ 94 Tokyo*, 7-18, 1994

[COWI96] J. Cowie and W. Lehnert, Information Extraction, *CACM* 39 (1), 80-91, 1996

[GOH97] C.H. Goh, S. Bressan, S.E. Madnick, and M.D. Siegel, The Context Interchange Approach to Semantic Interoperability, *submitted for publication*

[GRIS95] R. Grishman and B. Sundhiem, Design of the MUC-6 Evaluation, *Proc. 6th Message Understanding Conference (MUC-6)*, 1-11, 1995

[HARM95] D. Harmon (ed), *Proc. 3rd Text Retrieval Conference*, National Institute of Standards, Maryland, 1995

[LEWI96] D.D. Lewis, R.E. Shapire, J.P. Callan and R. Papka, Training Algorithms for Linear Text Classifiers, *Proc. ACM SIGIR Conf.*, 298-315, 1996

[LESJ96] D.D. Lewis and K. Sparck Jones, Natural Language Processing for Information Retrieval, *CACM*, 39 (1), 92-101, 1996

[LEVY96] A.L. Levy, A. Rajamaran and J.J. Ordille, Querying Heterogeneous Information Sources Using Source Descriptions, *Proc. 22nd Int. Conf. on VLDB Bombay*, 1996

[MEND96] A.O. Mendelzon, G.A. Milhaila and T. Milo, Querying the World Wide Web, U. Toronto Tech. Rep., 1996

[RILO94] E. Riloff and W. Lehnert, Information Extraction as a Basis for High-Precision Text Classification, *ACM TOIS*, 12 (3), 296-333, 1994

[RILO95] E. Rillof and J. Shoen, Automatically Acquiring Conceptual Patterns Without an Annotated Corpus, *Proc. 3rd Workshop on Very Large Corpora*, 148-161, 1995

[SALT94] G. Salton and A. Singhal, Automatic Text Theme Generation and the Analysis of Text Structure, Cornell U. Technical Report TR 94-1438, 1994

[SALT93] G. Salton, J. Allen and C. Buckley, Approaches to Passage Retrieval in Full Text Information Systems, *Proc. SIGIR Conf.* , 49-58, 1993

[SODE95] S. Soderland, D. Fisher, J. Aseltine and W. Lehnert, CRYSTAL: Inducing a Conceptual Dictionary, *Proc. 14th Int. Joint Conf. on Artificial Intelligence*, 1314-1321, 1995

[TOMA96] A. Tomasic, L. Raschid and P. Valduriez, Scaling Heterogeneous Database and the Design of DISCO, *Proc. Int. Conf. on Dist. Comp. Sys., Hong Kong*, 1996.