Semantic Information Elicitation from Unstructured Medical Records

Massimo Ruffolo[1,3], Vittoria Cozza[2], Lorenzo Gallucci[1,2] Marco Manna[4], Mariarita Pizzonia[2]

[1] Exeura s.r.l.
[2] DEIS - Department of Electronics, Computer Science and Systems
[3] ICAR-CNR - Institute of High Performance Computing and Networking
of the Italian National Research Council
[4] Department of Mathematics
University of Calabria, 87036 Arcavacata di Rende (CS), Italy
e-mail: ruffolo@icar.cnr.it
e-mail: manna@mat.unical.it
e-mail: {cozza,pizzonia}@deis.unical.it
e-mail: {ruffolo,gallucci}@exeura.it
WWW home page: http://www.exeura.it

Abstract. Semantic elicitation of relevant information entities from semi- and unstructured documents is an important problem in many application fields. This paper describes HıLεXa system implementing a very powerful semantic approach to information extraction from semi- and unstructured documents obtained combining knowledge representation formalisms, like ontology languages, and two-dimensional languages exploiting a two-dimensional spatial representation of documents. The HıLεX system constitutes a new generation technology capable of capturing and eliciting relevant information regarding a specific domain. It is founded on OntoDLP, an extension of disjunctive logic programming for ontology representation and reasoning. In the HıLεX system the semantics of the information to be extracted is represented by using OntoDLP ontologies and the extraction patterns are expressed by means of regular and two-dimensional expressions. By converting the extraction patterns to OntoDLP reasoning modules, the HıLεX system can actually extract information from HTML pages as well as from flat text documents using the same patterns. In this paper the extraction of clinical information and events, regarding patients, diseases, therapies and drugs, from electronic textual medical records is shown. Extracted information are represented in XML and can be stored in structured form using relational database or ad-hoc ontologies to enable further analysis.

1 Introduction

To extract automatically relevant information entities from unstructured electronic sources is an important problem of information management in many application fields. Information contained in semi- and unstructured documents, is usually arranged according to syntactic, semantic and presentation rules of a given natural language. While this is useful to humans, the lack of machine readability make impossible to manage the huge amount of information available on the Internet and in the document repositories of all kind of organizations. Information extraction allows the acquisition of information from semi- and unstructured documents and their storage in a structured machine-readable form useful for further analysis and exchanges. Existing information extraction systems are unable to handle the actual knowledge that the information conveys because the lack of semantic-awareness [2,1,11,13,7]. Being able to exploit the semantics is extremely important in order to recognize and extract automatically relevant information (entities) from semi- and unstructured documents.

In this paper is presented the application of HıLεX, a language independent system for semantic information extraction, to the extraction of information about clinical processes (regarding entities like patients, diseases, treatments, drugs, therapies, etc.) from electronic medical records having a flat textual form [8]. Extracted information are represented in XML and can be stored in relational database or in OntoDLP ontologies.

The system implements a logic based approach to information extraction which combines both syntactic and semantic knowledge for the expression of very powerful extraction patterns. In particular, the paper shows the extraction of information from a set of clinical records, in Italian language, regarding patients affected by lungs cancer.

The paper is organized as follow: in section 2 are described the HıLεX system, the ontologies and the patterns defined to extract medical entities of interest; in section 3 are shown the electronic medical records in input to the HıLεX system and the organization of the structured records obtained as output of the extraction process.

2 HıLεX system overview

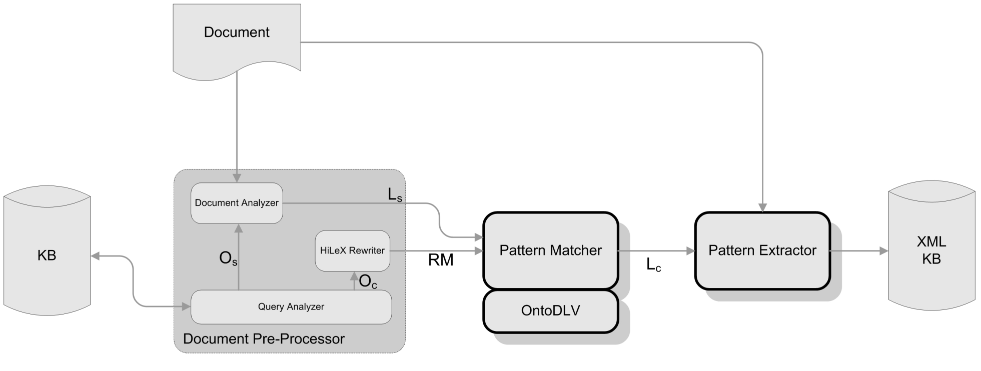


Fig. 1. HıLεX architecture

The HıLεX system is based on the exploitation of the OntoDLP [4] a powerful logic-based ontology representation language which extends Disjunctive Logic Programming (DLP) [9] with object-oriented features, such as relations, classes, object instances. Also, notions coming from object-oriented world are present, such as complex-objects, (multiple) inheritance and the concept of modular programming (by means of *reasoning modules*). This makes OntoDLP a complete ontology representation language supporting sophisticated reasoning capabilities. The OntoDLP language is implemented in the OntoDLV system, a cross-platform development environment for knowledge modeling and advanced knowledge-based reasoning. The OntoDLV system [4] permits to easily develop real world complex applications and allows advanced reasoning tasks in a user friendly visual environment. OntoDLP seamlessly integrates the DLV [6,12] system exploiting the power of a stable and efficient Answer Set Programming solver (for further background on DLV and DLP^{+}see [4,5]).

OntoDLP allows the formal representation of the semantics of information to be extracted (by means of suitable ontologies) and the encoding of the *logic two-dimensional representation* of unstructured documents. Moreover, OntoDLP reasoning modules (which are specialized OntoDLP logic programs) allow the exploitation of the the bottom-up reasoning capability, and thus the implementation of the logic-based pattern matching method yielding the actual semantic information extraction.

The semantic information extraction approach, implemented in the `HıLεX` system, can be viewed as a process composed of four main steps: knowledge representation, document preprocessing, pattern matching and pattern extraction. To understand how `HıLεX` works in the following the main system modules are shortly described. A more detailed description of the HiLEx system is given in [14].

2.1 Knowledge Base

The Knowledge Base (KB) stores core and domain ontologies describing the semantics of information to extract, extraction patterns and the logic *two-dimensional representation* of semi- and unstructured documents. The KB provide an API containing methods aimed at handling ontology querying and at assisting pattern specification and matching.

The elements of information to be extracted (entities) are modeled starting from the OntoDLP class *element* which is defined as follows:

```
class element (type: expression_type, expression: string,
   label: string).
```

The three attributes have the following meaning:

- `expression`: holds a string representing the pattern specified by regular expressions or by the `HıLεX` two-dimensional language, according to the `type` property. Patterns contained in these attributes are used to recognize the information elements in a document.
- `type`: defines the type of the expression (i.e. `regexp_type`, `hilex_type`).
- `label`: contains a description of the element in natural language.

The element class is the common root of both `core ontology` and `domain ontologies`. Every pattern encoding information to be extracted is represented by an instance of a class belonging to these ontologies.

The internal representation of extraction patterns is obtained by means of a two-dimensional language, founded on picture languages [3,10], and allowing the definition of very expressive target patterns. Each pattern represents a two-dimensional composition of portions annotated w.r.t. the elements defined in the ontology.

The two-dimensional language exploit the two-dimensional representation of an unstructured document, constituting the main notion, which the semantic information extraction approach implemented in the HıLεX system, is founded on. Following this idea elements are located inside rectangular regions of the input document called `portions` each univocally identified through the Cartesian coordinates of two opposite vertices. Document portions, and the enclosed elements, are represented in OntoDLP by using the class *point* and the relation *portion*

```
class point (x: integer, y: integer).
relation portion (p: point, q: point, elem: element).
```

Each instance of the relation `portion` represents the relative rectangular document region. It relates the two points identifying the region, expressed as instances of the class `point`, and an ontology element, expressed as instance of the class `element`. The set of instances of the `portion` relation constitute the *logic two-dimensional representation* of an unstructured document. This OntoDLP encoding allows to exploit the two-dimensional document representation for pattern matching.

In the following the structure of core and domain ontologies are described in greater detail.

The Core Ontology The core ontology represents general information elements valid in all the possible application domain. It is composed of three parts. The first part represents general elements describing a language (like, e.g., alphabet symbols, Part-of-Speech, regular forms such as date, e-mail, etc.). The second part represents elements describing presentation styles (like, e.g., font types, font styles, font colors, background colors, etc.). The third part represents structural elements describing tabular and textual structures (e.g. table cells, table columns, table rows, paragraphs, item lists, texture images, text lines, etc.). The core ontology is organized in the class hierarchy shown below:

```
class linguistic_element isa {element}.
    class character isa {linguistic_element}.
        class number_character isa {character}.
        ...
    class regular_form isa {linguistic_element}.
        class float_number isa {regular_form}.
        ...
    class separator isa {linguistic_element}.
    ...
class presentation_element isa {element}.
    class font_type isa {presentation_element}.
    ...
class structural_element isa {element}.
```

```
        class table_cell isa {structural_element}.
        ...
```

Examples of instances of these classes are:

```
    float_number: number (type: regexp_type,
                              expression:"(\\d{1,3}(?>.\\d{3})*,\\d+)").
```

When in a document a regular expression (pattern) characterizing a particular concept is recognized, a document portion is generated and annotated w.r.t. the corresponding class instance.

Domain Ontologies The Domain ontologies contain information elements of a specific knowledge domain. The distinction between core and domain ontologies allows to describe knowledge in a modular way. When a user need to extract data from a document regarding a specific domain, he can use only the corresponding domain ontology. The modularization improve the extraction process in terms of precision and overall performances.

A strong research effort has been taken, in the recent past, to provide an uniform representation of medical knowledge useful in health care information systems. Interesting results have been obtained in the field of medical knowledge representation, where many ontologies, such as *UMLS*, *ICD9-CM*, have been developed on different medical topics. In this work a domain ontology, inspired to *ICD9-CM*, has been implemented in OntoDLP to model knowledge and patterns about oncological domain. In the following is shown a piece of the ontology regarding the care of lung's cancer.

```
class cura_tumore_domain_element isa {tumore_domain_element}.
    class modalita isa {cura_tumore_domain_element}.
        class modalita_terapia isa {modalita}.
    class terapia_domain_element isa {cura_tumore_domain_element}.
        class chemioterapia isa {terapia_domain_element}.
        class radioterapia isa {terapia_domain_element}.
        class intervento_domain_element isa {terapia_domain_element}.
            class intervento_chirurgico isa {intervento_domain_element}.
    class medicinale_domain_element isa {cura_tumore_domain_element}.
        class farmaco isa {medicinale_domain_element}.
            class farmaco_chemioterapico isa {farmaco}.
        class posologia isa {medicinale_domain_element}.
```

The medical domain ontology that deals with therapies for the lung's cancer is shown graphically in figure 2. In particular, in the ontology are represented patterns expressing the three main modalities of lung's cancer care (therapies) the surgery, the x-ray and the chemotherapy. The HıLεX patterns allowing the extraction of chemotherapy (drugs dosage), expressed by using the construct of HıLεXtwo-dimensional language, *sequenceOf* are showed below:

```
farmaco_001: farmaco_chemioterapico (type: regexp_type,
```

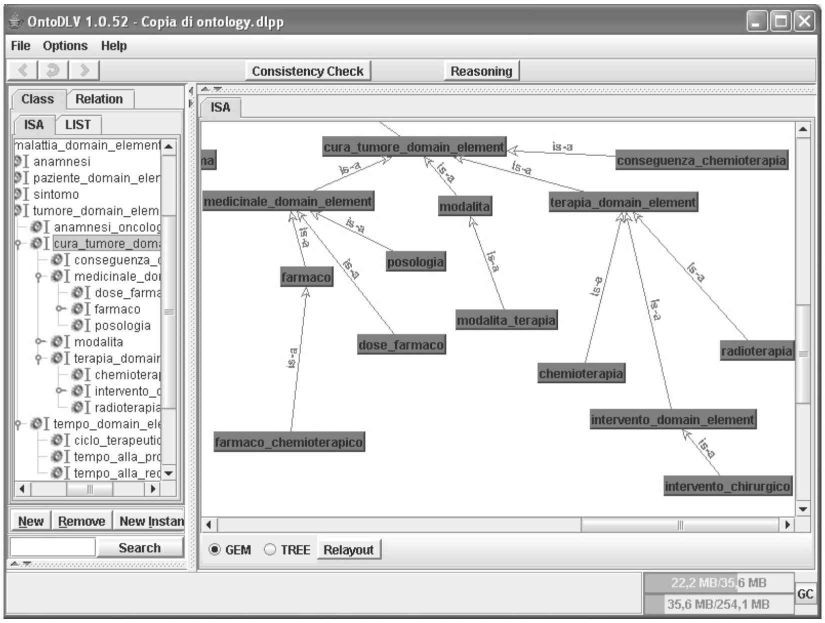


Fig. 2. Ontology snapshot

```
    expression: "cddp|cisplatino", label: " ").
farmaco_002: farmaco_chemioterapico (type: regexp_type,
    expression:"gemcitabina|gem|gemzar", label: " ").

posologia_01: posologia (type: hilex_type,
    expression:"sequenceOf(arg: [@number, @unita_misura, @number,
                @periodo], dir:horizontal, sep: sep002)", label: " ").

chemioterapia_01: chemioterapia (type: hilex_type,
    expression:"sequenceOf (arg: [@farmaco_chemioterapico,
                @posologia], dir: horizontal, sep: sep002)", label: " ").
```

The *sep002* instance define a separators among concepts. It helps to easily express that one or more white space can be found between the concepts *farmaco_chemioterapico* and *posologia*. It is worthwhile noting that each pattern allows to obtain a more complex concept as a two-dimensional composition of more simple information elements expressed by means of ontology concepts. The notation *@farmaco_chemioterapico* expresses a generic instance of the concept *farmaco_chemioterapico* that represents a set of possible chemotherapy drugs.

2.2 Document Preprocessor

The document preprocessor takes as input an unstructured document (i.e. the flat text medical records), and a set of class and instance names representing the information that the user wishes to extract. After the execution the document preprocessor returns the *two-dimensional logic representation* of the document and a set of reasoning modules constituting the input for the pattern matcher. The document preprocessing is performed by the three sub-modules described in the following.

Query analyzer. This submodule takes as input a set of classes and instances and explores the ontology in order to identify patterns for the extraction process. The output of the query analyzer are two sets of couples (*class instance name*, *pattern*). The first set (O_s) contains couples in which instances are characterized by patterns represented by regular expressions (simple elements), whereas in the second set (O_c) patterns are expressed using the HıLεX pattern representation language (complex elements). The set O_s is the input for the document analyzer submodule and the set O_c is the input for the rewriter submodule.

Document Analyzer. The input of this submodule is an unstructured document and the set of couples O_s. The document analyzer is able to recognize regular expressions, applying pattern matching mechanisms, to detect simple elements constituting the document and for each of them generates the relative *portion*. At the end of the analysis this module provides the *logic two-dimensional document representation* L_s which is a uniform abstract view of different document formats. In order to perform adequately even on large input documents the Document Analyzer is built as a reconfigurable set of specialized *processing units*, managed by a framework named CPF (Concurrent Processing Framework). Each unit is highly focused on a small part of the analysis process (e.g. a regular expression recognizer takes document fragments as input and can produce objects named *matches*), works on its input set members one at a time and is able to yield same results not dependant on the particular permutation of members sequence seen as input (i.e. a processing unit can work on its input set in any order). Since unit operation scheduling can be changed freely, while keeping correct results, the CPF is allowed to employ an execution strategy customized for a particular execution environment, in order to achieve top performances.

HıLεX **Rewriter.** The input for this submodule is the set of couples O_c containing the extraction patterns expressed by means of the HıLεX two-dimensional language. Each pattern is translated in a set of logical rules implemented in a OntoDLP reasoning modules (RM) which are to be executed by the OntoDLV system. The translation allows the actual semantic information extraction from unstructured documents performed by the pattern matcher module.

2.3 Pattern Matcher

The pattern matcher is founded on the OntoDLV system. It takes as input the logic two-dimensional document representation (L_s) and the set of reasoning modules (RM) containing the translation of the HıLεX patterns in term of logic rules and recognize new complex elements. The output of this step is the *augmented logic* *two-dimensional representation* (L_c) of an unstructured document in which new document regions, containing more complex elements (e.g. phrases containing information about chemotherapy, diagnosis, etc.) are identified. The logic-based pattern matching mechanism implemented in this module exploits the translation of extraction patterns performed by the HıLεX rewriter submodule.

It is noteworthy that patterns are very synthetic and expressive. Moreover, patterns are general in the sense that they are independent from the document format. This last peculiarity implies that the extraction patterns, presented above, are more robust w.r.t. variations of the page structure than extraction patterns defined in the previous approaches.

2.4 Pattern Extractor

This module takes in input the augmented logic representation of a document (L_c) and allows the acquisition of requested information entities. Acquired entities are represented in XML and can be stored in a OntoDLV ontology, a relational database, an XML database. So, extracted information can be used in other applications and more powerful query and reasoning task are possible on them. The extraction process causes the annotation of the documents w.r.t. the ontologies concepts. This feature can enable, for example in document management contexts the semantic classification.

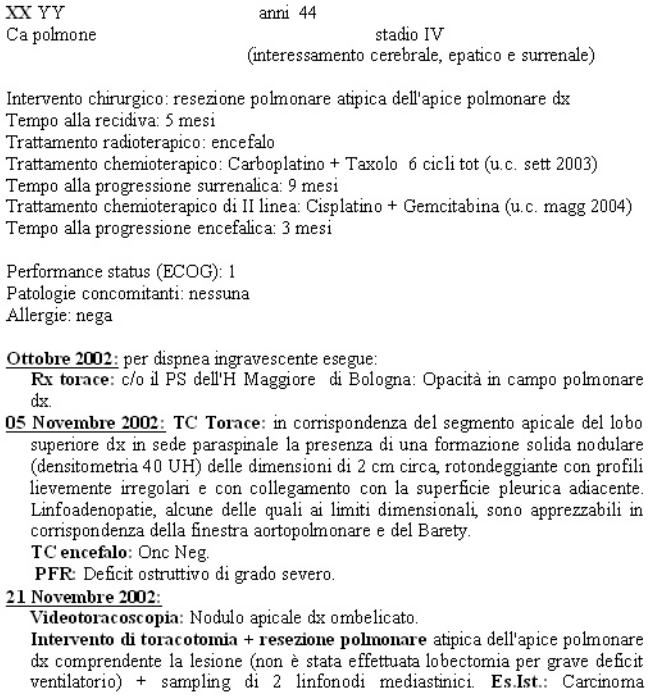
3 Information Extraction from unstructured medical records

In this section the clinical data extraction problem, from electronical medical records (EMR) in textual format, and the data structuring by means of XML, is faced. In the experiments has been extracted entities regarding hospital and ward name; patient identifier, sex an age; diagnosis and its date; oncological familiar analysis; chemotherapy; time to tumor progression and tumor recurrence; side effects of the chemotherapy. In the experiments has been used 100 EMR, written in Italian language, belonging to patients with lung's cancer. A cross-validation to proof the efficiency of the information extraction approach as been performed on them. In figure 3 is shown an EMR piece.

EMRs are weakly-structured documents (having usually 3 pages) since can be find in them, frequently, a standard structure. For example, the personal data of the patient are in the top of the document, the timetable of clinical events (medical exams, surgical operations, diagnosis, chemotherapy, etc.) is introduced by a date, and so on.

The semantic information extraction form the EMR is required because it's quite important for the doctors to have, easily and rapidly, information on the state of a patient, diagnosis, surgical operations and chemotherapies. These information should be available for all the medical staff in the same hospital, but also for more hospitals in different geographic areas.

The semantic of the medical information and the extraction patterns are represented as shown in the previous sections. The extracted entities are stored in a



XX YY anni 44

Ca polmone stadio IV

(interessamento cerebrale, epatico e surrenale)

Intervento chirurgico: resezione polmonare atipica dell'apice polmonare dx
Tempo alla recidiva: 5 mesi
Trattamento radioterapico: encefalo
Trattamento chemioterapico: Carboplatino + Taxolo 6 cicli tot (u.c. sett 2003)
Tempo alla progressione surrenalica: 9 mesi
Trattamento chemioterapico di II linea: Cisplatino + Gemcitabina (u.c. magg 2004)
Tempo alla progressione encefalica: 3 mesi

Performance status (ECOG): 1
Patologie concomitanti: nessuna
Allergie: nega

Ottobre 2002: per dispnea ingravescente esegue:
Rx torace: c/o il PS dell'H Maggiore di Bologna: Opacità in campo polmonare dx.

05 Novembre 2002: TC Torace: in corrispondenza del segmento apicale del lobo superiore dx in sede paraspinale la presenza di una formazione solida nodulare (densitometria 40 UH) delle dimensioni di 2 cm circa, rotondeggiante con profili lievemente irregolari e con collegamento con la superficie pleurica adiacente. Linfoadenopatie, alcune delle quali ai limiti dimensionali, sono apprezzabili in corrispondenza della finestra aortopolmonare e del Barety.
TC encefalo: Onc Neg.
PFR: Deficit ostruttivo di grado severo.

21 Novembre 2002:
Videotoracoscopia: Nodulo apicale dx ombelicato.
Intervento di toracotomia + resezione polmonare atipica dell'apice polmonare dx comprendente la lesione (non è stata effettuata lobectomia per grave deficit ventilatorio) + sampling di 2 linfonodi mediastinici. **Es.Ist.:** Carcinoma

Fig. 3. Input: clinical record (flat text)

common and understandable XML format. A piece of XML record in output from the extraction process is the XML element presented below:

```
<chemioterapia>
    <patient_id> 8723746 <patient_id>
    <date> 13/01/2006 </date>
    <farmaco>
        <name> Platinex </name>
        <posologia> 75 mg/mq 1 g  </posologia>
    </farmaco>
    <farmaco>
        <name> Gemzar </name>
        <posologia> 1200 mg/mq 1,8 gg </posologia>
    </farmaco>
</chemioterapia>
```

In order to extract the type of surgery on the patient, complex operations, for example the removal or the resection of a particular part of the body, implemented through the composition of lower level HιLεX expressions, have been expressed. To extract the item related with x-ray, after expressing the concept of x-ray and several ways to express it, it has been taken into consideration the way to execute such therapy or the purpose of such therapy (palliative or analgesic scope).

4 Conclusions and future works

The semantic approach to information extraction presented in this work is novel, powerful and expressive, as well as concrete (it is implemented in the HıLεX system), and constitutes an enhancement in the field of information extraction. Unlike previous approaches, the same extraction patterns can be used to extract information, according to their semantics, from different kind of semi- and unstructured documents (HTML, flat text).

This work shows, also, how semantic information extraction allows the acquisition of relevant entities of a domain. Using HıLεX doctors can acquire and structure information about clinical process without change in the usual clinical practices. The definition of suitable extraction patterns allows to obtain main EMR information when it are written in flat text format. In particular, information about the kind of patient surgery under some particular conditions (age, familiar oncological anamnesis), the therapy for the patient, the effects on the patient because such therapy, the surgical operations and its results, the time a disease takes to propagate and finally the time the cancer takes to develop itself again after the last surgery, can be captured.

Using the obtained structured data (a structured EMR for each patient) the oncological ward of the hospital can monitor the state of disease of patients daily. The structured EMR can also be used to exchange information on the patient with others clinical center where, for example, patient could be hosted in the future. Moreover, obtained EMR can be exchanged among medical researchers to try discovering, for example by means of data mining methods, the effect of innovative diagnostic approaches and/or therapeutic procedures and the adverse reactions to some drugs.

Currently, consolidation of the approach is ongoing and its theoretical foundations are under investigation and improvement. Future work will be focused on the consolidation and extension of the HıLεX two-dimensional language, the investigation of the computational complexity issues from a theoretical point of view, the extension of the approach to PDF as well as other document formats, the exploitation of natural language processing techniques aimed at improve information extraction from flat text documents.

References

1. R. Baumgartner, S. Flesca, and G. Gottlob. Declarative information extraction, web crawling, and recursive wrapping with lixto. In *LPNMR '01: Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning*, pages 21–41, London, UK, 2001. Springer-Verlag.
2. R. Baumgartner, S. Flesca, and G. Gottlob. Visual web information extraction with lixto. In *The VLDB Journal*, pages 119–128, 2001.
3. S.-K. Chang. The analysis of two-dimensional patterns using picture processing grammars. In *STOC '70: Proceedings of the second annual ACM symposium on Theory of computing*, pages 206–216, New York, NY, USA, 1970. ACM Press.
4. T. Dell'Armi, N. Leone, and F. Ricca. Il linguaggio dlp+. Internal report, Exeura s.r.l, June 2004.

5. T. Eiter, W. Faber, N. Leone, and G. Pfeifer. Declarative Problem-Solving Using the DLV System. In J. Minker, editor, *Logic-Based Artificial Intelligence*, pages 79–103. Kluwer Academic Publishers, 2000.
6. W. Faber and G. Pfeifer. Dlv homepage, since 1996.
7. R. Feldman, Y. Aumann, M. Finkelstein-Landau, E. Hurvitz, Y. Regev, and A. Yaroshevich. A comparative study of information extraction strategies. In A. F. Gelbukh, editor, *CICLing*, volume 2276 of *Lecture Notes in Computer Science*, pages 349–359. Springer, 2002.
8. C. Friedman, G. Hripcsak, W. DuMouchel, S. B. Johnson, and P. D. Clayton. Natural language processing in an operational clinical environment. In *Natural Language Engineering*, volume 1, pages 83–108, 1995.
9. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3/4):365–386, 1991.
10. D. Giammarresi and A. Restivo. Two-dimensional languages. In A. Salomaa and G. Rozenberg, editors, *Handbook of Formal Languages*, volume 3, Beyond Words, pages 215–267. Springer-Verlag, Berlin, 1997.
11. S. Kuhlins and R. Tredwell. Toolkits for generating wrappers – a survey of software toolkits for automated data extraction from web sites. In M. Aksit, M. Mezini, and R. Unland, editors, *Objects, Components, Architectures, Services, and Applications for a Networked World*, volume 2591 of *Lecture Notes in Computer Science (LNCS)*, pages 184–198, Berlin, Oct. 2003. International Conference NetObjectDays, NODe 2002, Erfurt, Germany, October 7–10, 2002, Springer.
12. N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, and F. Scarcello. The DLV System for Knowledge Representation and Reasoning. 2004.
13. B. Rosenfeld, R. Feldman, M. Fresko, J. Schler, and Y. Aumann. Teg: a hybrid approach to information extraction. In D. Grossman, L. Gravano, C. Zhai, O. Herzog, and D. A. Evans, editors, *CIKM*, pages 589–596. ACM, 2004.
14. M. Ruffolo, N. Leone, M. Manna, D. Sacc, and A. Zavatto. Exploiting asp for semantic information extraction. In *Answer Set Programming*, 2005.