Towards a Semantic Information Extraction Approach from Unstructured Documents

Massimo Ruffolo¹,³, Lorenzo Gallucci¹,² Nicola Leone¹,⁴ Marco Manna⁴ Domenico Saccà¹,²,³

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: {leone,manna}@mat.unical.it
e-mail: {sacca}@deis.unical.it
e-mail: {ruffolo,gallucci,leone,sacca}@exeura.it
WWW home page: http://www.exeura.it

Abstract. Recognizing and extracting meaningful information from semi- and unstructured documents, taking into account their semantics, and storing them into a database is an important problem in the context of information access and retrieval. This paper describes a novel logic-based approach to information extraction from both semi- and unstructured documents. The approach, implemented in the HIX system, is founded on a new two-dimensional representation of documents constituting a unified abstract representation of both HTML pages and flat text documents. The semantics of information to be extracted is encoded by means of ontology expressed in $DLP^+$, - an extension of disjunctive logic programming for ontology representation and reasoning, which has been recently implemented on top of the DLV system. Unlike previous systems, which are mainly syntactic, HIX combines both semantic and syntactic knowledge for a powerful information extraction. Each extraction pattern belongs to an ontology class and is expressed using regular expressions and/or an ad hoc two-dimensional language exploiting the document two-dimensional representation. The execution of $DLP^+$ reasoning modules, encoding the HIX language expressions in terms of logic rules, yields the actual extraction of information from the input document. HIX allows the semantic information extraction from both HTML pages and flat text documents by using synthetic and very expressive extraction patterns.

1 Introduction

HTML and flat text documents contain a huge amount of information arranged for human readers according to syntactic, semantic and presentation rules of a given language. That information tends to be practically unusable, both for their vastness, and the lack of machine readability that makes existing information extraction systems unable to manage the actual knowledge that the information conveys.

In this work is presented HIX, a logic-based system combining both syntactic and semantic knowledge for a powerful information extraction from semi- and unstructured documents. Extracted information can be stored into XML or relational databases.
In the recent literature a number of approaches for information extraction from unstructured documents have been proposed. An overview of the large body of existing literature and systems is given in \[5,9,12,13,15\]. The currently developed systems are mainly syntactic, and are not aware of the semantics of the information they are able to extract. They principally use pattern matching mechanism exploiting the underlying HTML syntactical structure and regular expressions on textual fragments contained between HTML tags.

Logic-based approaches for information extraction are not new \[1,2\], however, the approach we propose is original. Its novelty is due to:

- The two-dimensional representation of semi- and unstructured document. A document is viewed as a Cartesian plane composed by a set of nested rectangular regions called portions. Each portion, univocally identified through the cartesian coordinates of two opposite vertices, contains a piece of the input document (element) annotated w.r.t. an ontology. The two-dimensional documents representation provides a unified abstract representation of HTML and flat text, so the same extraction patterns can be used on different document format.

- The exploitation of a logic-based knowledge representation language called \(DLP^+\), extending \(DLP\) \[10\] with object-oriented features, including classes, (multiple) inheritance, complex objects, types. \(DLP^+\) is well-suited for representation and powerful reasoning on ontologies; the language is supported by the \(DLV^+\) system \[4\], implemented on top of \(DLV\) \[6,7,8,14\].

- The use of ontologies, encoded in \(DLP^+\), describing the domain of the input document. A concept of the domain is represented by a \(DLP^+\) class; each class instance is a extraction pattern representing a possible way of writing the concept and is used to recognize and annotate an element.

- The employment of a new language, named \(HL\kappa X\) two-dimensional language, for specifying the (above mentioned) extraction patterns. \(HL\kappa X\) language extends regular expressions for the representation of two-dimensional patterns (like tables, item lists, etc.), which often occur in web pages and textual tabular data. The patterns are specified through \(DLP^+\) rules, whose execution yields the actual semantic information extraction, by associating an element (the part of the document embraced by) each portion to a class of the ontology.

The advantages of the \(HL\kappa X\) system over other existing approaches are mainly the following:

- The extraction of information according to their semantics and not only on the basis of their syntactic structure.

- The possibility to extract information in the same way from documents in different formats. The same extraction pattern can be used to extract data from both flat text and HTML documents.

- The possibility to obtain a “semantic” classification of the input documents w.r.t. an ontology, which is much more accurate and meaningful than the classifications provided by existing systems (mainly based on statistical approaches exploiting the occurrences of some relevant words). This feature opens the door to many relevant applications (e.g., emails classification and filtering, skills classification from curricula, extraction of relevant information from medical records, etc.).

Distinctive features of the \(HL\kappa X\) system, summarized above, allow a better digital contents management and fruition in different information technology areas such as: competitive and information intelligence, document and content management for e-health, e-entertainment, e-commerce, e-government, e-business applications.
This paper propose a by example explanation of the semantic information extraction approach implemented in the HiLeX system. The example concerns the extraction of information about the stock market indexes from a table contained in a semi-structured Web page and in a flat text document. The remainder of this work is organized as follows. Section 2 shows the knowledge representation mechanisms, section 3 shows the logic two-dimensional document representation idea, section 4 describes how extraction patterns are represented, section 5 is focused on the logic-based pattern matching method (performed by $DLP^+$ reasoning modules, finally section 6 gives an overview of the HiLeX system.

2 Knowledge Representation

In the HiLeX system the information to be extracted are modeled by using two kind of ontologies, the core ontology and the domain ontologies. Both core and domain ontologies are represented using $DLP^+$ [4] a powerful logic-based language which extends Disjunctive Logic Programming (DLP) [6] by object-oriented features. In particular, the language includes, besides the concept of relations, other notions coming from object-oriented world such as classes, objects (class instances), object-identity, complex-objects, (multiple) inheritance, and the concept of modular programming by means of reasoning modules. This makes $DLP^+$ a complete ontology representation language supporting sophisticated reasoning capabilities. Core and domain ontologies have a common root class called $element$ which $DLP^+$ encoding is shown in the follows:

\[
\text{class element (type: expression_type, expression: string, label: string).}
\]

The three attributes have the following meaning:

- **expression**: holds a string representing the pattern specified by regular expressions or by the HiLeX two-dimensional language, according to the type property. Patterns contained in these attributes are used to recognize the 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. In the following the structure of core and domain ontologies are described in detail.

2.1 Core Ontology

The core ontology is composed of three parts. The first part represents general simple elements describing a language (like, e.g., alphabet symbols, lemmas, 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:

\[
\text{class linguistic_element isa \{element\}.}
\]
\[
\text{class character isa \{linguistic_element\}.}
\]
\[
\text{class number_character isa \{character\}.}
\]
...\[
\text{class regular_form isa \{linguistic_element\}.}
\]
\[
\text{class float_number isa \{regular_form\}.}
\]
...
2.2 Domain Ontologies

Domain ontologies contain simple and complex 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. Referring to the example of figure 1 and 2, elements representing concepts related to the stock index market domain can be organized as follows:

```plaintext
class stock_market_domain isa {element}.
class stock_index isa {stock_market_domain, linguistic_element}.
class stock_index_cell isa {stock_market_domain, structural_element}.
class stock_index_row isa {stock_market_domain, structural_element}.
class stock_index_table isa {stock_market_domain, structural_element}.
class index_value isa {stock_market_domain, regular_form}.
```

3 Logic Two-Dimensional Representation of Documents

The two-dimensional representation of semi- and unstructured document is the main notion, which the semantic information extraction approach implemented in the HILX system, is based on. This notion is founded on the idea that semi- and unstructured documents can be considered as a Cartesian plane composed by a set of nested rectangular regions called portions. Each portion, univocally identified through the Cartesian
coordinates of two opposite vertices, contains a piece of the input document representing an element of the information to be extracted (figure 2). Document portions, and the enclosed elements, are represented in $DLP^+$ by using the class point and the relation portion shown below:

```plaintext
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 $DLP^+$ encoding allows to exploit the two-dimensional document representation for the information extraction process.

### 4 Extraction Patterns Specification

Every extraction pattern, encoding information element to be extracted, is represented by an instance of a class belonging to core or domain ontologies.

The representation of extraction patterns is obtained by means of regular expressions or by the HileX two-dimensional language, founded on picture languages \cite{3,11}, and allowing the definition of very expressive target patterns. Patterns expressed using HileX two-dimensional language represent a two-dimensional composition of portions annotated w.r.t. elements defined in the ontology. Examples of instances of the english_lemma, float_number and stock_index classes, containing pattern expressed by regular expressions, are:

```plaintext
/*1*/ system: english_lemma (type: regexp_type, expression: "system").
/*2*/ unsigned_float_number: float_number (type: regexp_type,
expression: "(\d{1,3}(?>\.*\d{3})*,\d+)", label: "...").
/*3*/ mibtel: stock_index (type: regexp_type, expression: "Mibtel").
```

When in a document the regular expression characterizing an English lemma, an unsigned float number, a stock index is recognized, a document portion is generated and annotated w.r.t. the corresponding class instance.

An extraction pattern, contained in an instance of the stock_index_table class, obtained by using the sequenceTableOf HileX grammar construct as shown in the following:

```plaintext
stock_index_table_01:stock_index_table( type: hilex_type,
expression:"sequenceTableOf( arg: [stock_index, absolute_index_value,
absolute_index_variation, percentage_index_variation],
range:<2,5>, dir:vertical, sep:sep_01 "),
label:"table containing stock_index_row").
```

The instance stock_index_table_01 represents a table of stock index variations composed by a vertical sequence of at least 2 and at most 5 rows. Each row is a sequence of other portions annotated w.r.t. instances of the class stock_index, and the unsigned_float, signed_float and a percentage (i.e. a stock_index_row) instances.
5 Logic-Based Pattern Matching

The pattern matching mechanism is implemented encoding patterns expressed by means of the HiLeX two-dimensional language in \(DLP^+\). In particular, each pattern is rewritten in a \(DLP^+\) reasoning module as a set of rules exploiting basic operators able to manipulate points and portions. The table containing the stock index variations, incorporated in the page shown in figure 1, can be extracted using the pattern presented in the section 4. The corresponding \(DLP^+\) rewriting is shown below.

\[
\text{module(stock_index_table_01) \{} \\
\quad \text{portion}(P_1, Q_7, \text{stock_index_table_row_01}):\text{-} \\
\quad \text{E1:stock_index()}, \text{strictFollow}(P_1, Q_1, E_1, P_2, Q_2, \text{sep}_01), \\
\quad \text{strictFollow}(P_2, Q_2, \text{sep}_01, P_3, Q_3, \text{absolute_index_value}), \\
\quad \text{strictFollow}(P_3, Q_3, \text{absolute_index_value}, P_4, Q_4, \text{sep}_01), \\
\quad \text{strictFollow}(P_4, Q_4, \text{sep}_01, P_5, Q_5, \text{absolute_index_variation}), \\
\quad \text{strictFollow}(P_5, Q_5, \text{absolute_index_variation}, P_6, Q_6, \text{sep}_01), \\
\quad \text{strictFollow}(P_6, Q_6, \text{sep}_01, P_7, Q_7, \text{percentage_index_variation}). \\
\quad \text{portion}(P, Q, \text{stock_index_table_01}):\text{-} \\
\quad \text{verticalRecurrence}(P, Q, \text{stock_index_table_row_01}, \text{min:2}, \text{max:5}). \\
\}\n\]

The new portion having the semantic of table of stock index variation, whose structure satisfies the extraction pattern, is recognized by applying rules contained in the reasoning module shown above. It is noteworthy that patterns are independent from the document format. This 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. For example, if table containing the stock index variations is moved to another location in the page the extraction pattern mustn’t be modified. Furthermore, the same extraction patterns can be also used to extract the same information from document in which the stock index table is represented in flat text form.

This approach allows, also, to work with flat text document to recognize phrases having a particular meaning. For example, consider the following text:

"While people have criticized Knowledge Management as the same old thing using different words, a well-developed knowledge management system would stimulate the creativity ..."

Suppose to be interested on the recognition of the concept of "knowledge management system". In figure 3 is depicted the two-dimensional representation of the piece of the flat text, containing the concept of "knowledge management system", obtained after the preprocessing.

![Fig. 3. Example of portions in flat text](image)

Writing the following pattern:

\[
\text{knowledge_management_system_01: knowledge_management_system (type: hilex_type, expression: "sequenceOf(arg: [knowledge, management,system], dir: horizontal, sep: sep_01)", label:"..."}).
\]
the concept of knowledge management system can be recognized in all the flat text document in which it is present. This enable semantic search, semantic text annotation, automatic meta-data extraction.

6 HıLeX System Overview

The architecture of the HıLeX system is represented in figure 4. The Knowledge Base (KB) of HıLeX stores the core and domain ontologies by means of the DLV+ system persistency layer. The information extraction process is executed in three main steps: document pre-processing, pattern matching, and pattern extraction. Each step is performed by a suitable architectural module.

Fig. 4. The Architecture of the HıLeX System

In the first step a Document Pre-Processor takes in input an unstructured document and a set of class instances names, representing the information that the user needs to extract. Than the document preprocessor returns the two-dimensional logic document representation \( L_s \) and a set of reasoning modules, constituting the input for the pattern matcher (RM).

The Pattern Matching module is founded on the DLV+ system. It takes as input the logic document representation \( (L_s) \) and the set of reasoning modules (RM) and recognize new complex elements. The output of this step is the augmented logic representation \( (L_c) \) of a unstructured document in which new document regions, containing more complex elements are identified exploiting the semantic knowledge represented in the ontologies. The pattern matching is completely independent from the document format.

Finally, a Pattern Extractor takes in input the augmented logic representation of a document \( (L_c) \) and allows the acquisition of element instances (semantic wrapping) and/or the document classification w.r.t. the ontologies classes. Acquired instances can be stored in DLP+ ontologies, relational and/or XML databases.

7 Conclusions and Future Works

This work presents a novel, concrete, powerful and expressive approach to information extraction from semi- and unstructured documents grounded on two main ideas:

- The logic two-dimensional representation of documents allowing the definition of general and very expressive extraction patterns expressed by the HıLeX two-dimensional language.
- The expression of the semantics of the information to extract by means of the DLP+ ontology representation language, having solid theoretical foundations.
The approach constitutes a decisive enhancement in field of information access and extraction. Unlike previous approach, the same extraction patterns can be used to extract information, according to their semantics, form both HTML and flat text documents. Future work will be focused on the consolidation and extension of the \textsc{Hlhx} two-dimensional language, the investigation of computational complexity issues from a theoretical point of view, the extension of the approach to pdf and other document formats, the exploitation of natural language processing techniques aimed to improve information extraction from documents with only textual contents.

References