Purchasing the Web: an Agent based E-retail System with Multilingual Knowledge

Maria Teresa Pazienza, Armando Stellato, Michele Vindigni
DISP - University of Rome "Tor Vergata", Italy
{pazienza, stellato, vindigni}@info.uniroma2.it

Abstract

The more than enthusiastic success obtained by e-commerce and the continuous growth of the WWW has radically changed the way people look for and purchase commercial products. E-retail stores offer any sort of goods through evermore appealing web pages, sometimes even including their own search-engines to help customers find their loved products. With all this great mass of information available, people often get confused or simply do not want to spend time browsing the internet, loosing themselves into myriads of available e-shops and trying to compare their offers. E-retail systems are the natural solution to this problem: they place at people's disposal user-friendly interfaces, helping the customers in finding products from different e-shops that match their desires and comparing these offers for them.

Inside CROSSMARC, (a project funded by the Information Society Technologies Programme of the European Union: IST 2000-25366) different techniques coming from the worlds of NLP, Machine Learning-based Information Extraction and Knowledge Representation have been considered and conjoined to give life to an agent-based system for information extraction (IE) from web pages, which operates in a wide range of situations involving different languages and domains.

In this paper we describe the main components that realize the CROSSMARC architecture, together with their specific role in the process of extracting information from the web and presenting them to the user in a uniform and coherent way.

1. Introduction

Following the growing demand for user-friendly systems, dedicated to help people (not necessarily skilled with computers) solve everyday life tasks in an easier and convenient way, e-retail portals are becoming even more competitive than before: nowadays a wide variety of commercial agent-based systems currently guide many users in choosing online products, helping them to select features they may recognize as important for their needs and comparing on these basis the different product offers, in order to reach optimal satisfaction for the customer. In some cases, these agents simply access to a list of confederated sites which adhere to some standard in presenting their offers, in other cases, they have to mine relevant data from human-readable on-line product descriptions, extracting the information requested by the customer and presenting it in a synthetic and coherent way. Even those agents belonging to this latter category, typically don't use natural language technologies, and hence process strictly structured texts only, where product names, prices, and other features always appear in a fixed (or at least regular) order, making possible to use the page structure and/or mark-up tags as content delimiters.

Unfortunately, this kind of structured information is not what we expect to find in the web, where organization of web-pages usually tends more towards providing immediate human readability and giving emphasis on presentation of the products, than caring about how information can be easily extracted from automatic systems. Under this perspective, images and texts both contribute to the relevant information, being combined in a sometimes indivisible informational unit hard to disclose with ordinary web-mining techniques.

Things get even more complicated if we think about the possibility of examining and comparing offers from various countries, as we had to deal with different languages (and, consequently, with different character encodings used to represent their specific idioms); as a last consideration, technology and fashion push ahead very fast, seeing old concepts being unused and other ones emerging in the ever-evolving domains, moreover, even old recognized features may loose or gain importance in the aim of comparing products, or simply change the way we had to consider them (e.g. evaluation of prices or performances).

With this in mind, it's clear how standardization of the data structure that enclose the knowledge of a system, and strong decoupling of this data from the processing components, are necessary requisites to achieve optimal adaptivity towards different scenarios and applications.

We will describe here our contribution in building CROSSMARC, an e-retail product comparison multi-agent system, currently under development as part of an EU-funded project, aiming to provide users with product information fitting their needs. Inside CROSSMARC, technologies have been developed for extracting information from domain specific web pages, exploiting

language processing methods, machine learning techniques and a solid knowledge representation model in order to facilitate porting of the system to new domains. CROSSMARC also features localization methodologies and user modeling techniques in order to provide the results of extraction in accordance with the user's personal preferences and constraints.

2. System Architecture

The overall CROSSMARC architecture (see below Fig. 1) is realized through distributed but interoperable agents who communicate each other via a dedicated XML language in order to actuate, and coordinate at the same time, the different tasks that characterize the system's behavior.

Individual monolingual agents using XML to communicate each other have been plugged in: each partner in the project contributes with his autonomous agents, exchanging information through a common vocabulary provided by a domain specific ontology.

Agent roles in the architecture are primarily related to three main tasks:

- Implement a user interface, to process users' queries, perform user modeling, access the database and supply the user with product information.
- Extract Information from the WEB: here various processing steps are coordinated to find, retrieve, analyze and extract information from the Internet.
- Store the extracted information in a database, in order to feed the system with the data to be presented to the use

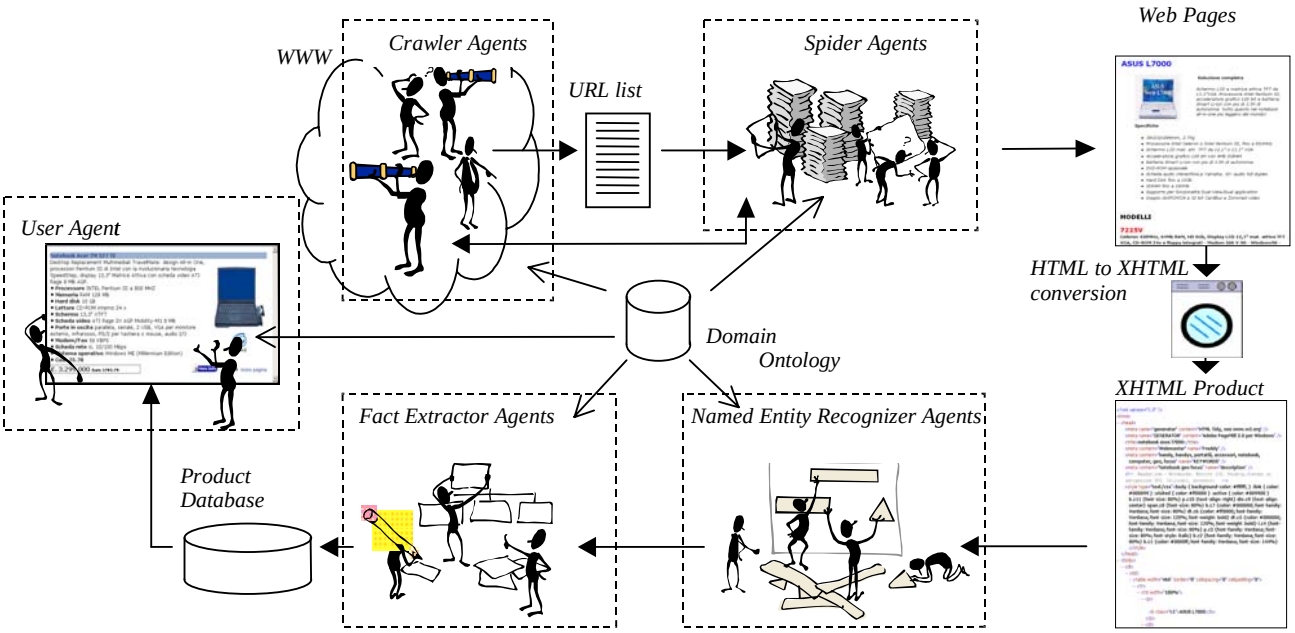

Fig. 1: Overall CROSSMARC Architecture

In particular, considering the main role of the Extraction agents, they are broadly divided into two categories, depending on their specific tasks:

- **Information retrieval agents (IR),** which identify domain-relevant Web Sites (*focused crawling*) and return web pages inside these sites (*web spidering*) that are likely to contain the desired information;
- **Information Extraction (IE) agents** (two separate agents for each language) which process the retrieved web pages. There are specific roles for every step of the extraction process: *Named Entity Recognition and Classification* (NERC), i.e. recognition of concepts pertaining to the domain of interest inside the web pages, *Fact Extraction* (FE), which consists in the identification of the number of products and how they are distributed in the web pages (*products demarcation*), and in the extraction of all the characteristics of these products.

All the agents share a common Knowledge model, which can easily be customized to new Domains, Languages and Extraction Templates. The customization process can be easily performed through an application (developed inside CROSSMARC) based on the APIs of the ontology editing tool Protégé-2000 [4], from the university of Stanford; an XML version of these ontologies and the FE XML Schema for every domain are then automatically derived from it through a specially designed plug-in that has been developed for this purpose inside the CROSSMARC Project.

Now we'll give more details regarding specific CROSSMARC components and how their work is coordinated to extract relevant information from the web.

3. Web Pages Collection

The process of collecting domain-specific web pages articulates in two different and complementary sub-processes:

- **focused crawling** to identify Web sites (e-retailers web sites) relevant to a specific domain (e.g. electronic products/computer goods)
- **domain-specific spidering of a Web site** to navigate through a specific Web site (e.g. retailer of electronic products), retrieving Web pages of interest (e.g. product descriptions).

Interesting Web sites are initially identified by an external focused crawling process. Then each site is spidered, starting at the top page, scoring the links in the page and following "useful" links.

4. Multilingual NERC and Name Matching

The Multilingual NERC subsystem architecture is shown in Fig. 2 where the individual components are autonomous agents, which need not to be installed all on the same machine.

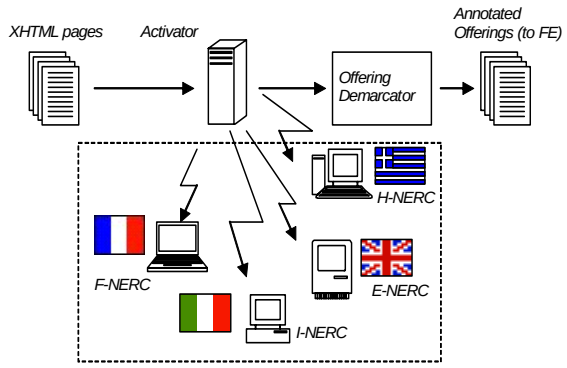

Fig. 2: architecture of the Named Entity Recognizer Component

We'll give now a inside view of the Italian Named Entity Recognizer and Classificator.

4.1 The Italian NERC

The I-NERC agent receives, from previous modules in the chain, the Web pages as XML structures containing description of one or more products from Italian retailers, then processes and enriches it by adding markup tags for domain specific information (i.e., product name, manufacturer, cost, etc.). In order to mark up relevant named entities, the I-NERC exploits two kinds of evidences:

- *Internal evidences*, that is information related to the entity itself.
- *External evidences* provided by the context in which the entity appears, which can in turn be divided into:
    i. *Structural evidences*, provided by the document organization (tables, bullet lists, or, at a lesser extent, formatting properties as bold, italic, etc.).
    ii. *Semantic evidences*, provided by its content (as in the case mentioned above).

The overall I-NERC agent is implemented as a sequence of processing steps driven by XSLT transformations over the XML input structure, by using a XSLT parser with a number of plugged-in specific extensions. A sequence of linguistic processes is activated applying a pipeline of transformations on the page

1) The **Normalizer** transformation provides pre-processing at character-level of the source document, to deal with bad word separators, wrong punctuation and word-level misspelling or ungrammaticalities.
2) The **Tokenizer** transformation applies to the page content, segmenting the text into atomic tokens, classified as word, number and separators and included in XML appropriate tags.
3) The **Terminology** transformation recognizes terminological expressions as well as simple constituents and expresses them in their standardized form.
4) **Lexicon-lookup** matches lexical rules and entries against the input. This phase relies on lexical knowledge, represented by an Italian lexicon, and additional lexical tables for specific information (for instance, measurement units).
5) **Unit-matcher** activates numerical expressions recognition in order to identify currencies, dates, lengths, and other domain specific quantities.
6) **Ontology-lookup** matches identified entities against the ontology and categorizes them accordingly.

5. Multilingual and Multimedia Fact Extraction

From a black-box point of view, the overall architecture of the Multilingual and Multimedia Fact Extraction (FE) component is analogous to the one of the NERC, being the input again represented by XHTML web pages, in this case enriched by NERC semantic annotations.

The first processing step performed by the Fact Extractor agents is *Product Demarcation*: NERC output is transformed by the *Product Demarcator module* (PD) of the Fact Extractor which analyzes the semantic categories identified by the NERC, tries to find

correlations between them and subsequently aggregates them into separate products.

Following these steps, the FE component exploits the NERC + PD annotations in order to identify which of the recognized semantic entities fill a specific fact slot inside a product description, according to the above mentioned XML FE Schema.

Inside CROSSMARC, different FE components have been realized from the four partners involved in the project. The common characteristic of all the Fact Extractor components is that all of them implement wrapper induction techniques for extracting only the analyzed information pertaining to the products recognized inside the pages. In particular a first version of the English Fact Extractor was based on Boosted Wrapper Induction [3], the Greek version of the Fact Extractor module is based on STALKER [1], while the Italian one is a customized implementation of the Whisk algorithm [2].

Obviously the scenario depicted is quite different from many of the typical wrapper induction approaches, in this case, strong semantic analysis performed by other linguistic processors modifies the search space of pure wrapper induction modules, limiting the number of valid extractions that a wrapper can make to those which maintain coherency with the structure of the pages and of the products presented inside them.

5.1 Italian Fact Extractor Component

As previously outlined, the Italian FE System has its core in a customized implementation of Soderland's WHISK algorithm of Wrapper Induction.

WHISK takes as input a set of hand-tagged instances, using them as a pool (the *training set*) for induction of IE rules, expressed in the form of regular expressions. These rules are induced top-down, first finding the most general rule that applies with success to the considered training instance, then producing new extended rules by adding terms one at a time and testing their behavior against the whole training set. The candidate extended rule that performs best against the whole training set is thereby chosen and examined for new possible extensions. The process is then reiterated until all the candidate extensions do not perform better than the rule produced in the previous step.

WHISK is capable of learning both single-slot and multi-slot rules, though we considered only single-slot rules because of the wide heterogeneity of combinations in which products can be presented: even in pages with similar structure or inside a single page, different products may vary in the number of characteristics they show or in the disposition of these characteristics among the description of the offer. For this reason, Italian FE relies on localization of the products inside the pages provided by the previous Product Demarcation component.

5.1.1 Whisk adaptation to CROSSMARC's needs

Soderland's original algorithm has been customized to meet the specific needs of the CROSSMARC environment, through the following aspects:

a) Ontology Lookup. WHISK uses the notion of Semantic Class to address disjunctive sets of terms that can be considered as equivalence classes. At the same time the concept of Semantic Tag is added to wrap concepts that may appear in a multitude of different aspects.

Both these two options have been adopted in CROSSMARC implementation of the WHISK algorithm, since all the NE tags (enriched by Product Demarcator attributes) are dynamically imported as Semantic Tags, while sets of Semantic Classes are defined on the basis of elementary concepts present in the ontologies.

b) Limiting Search Space of Induction when adding terms. WHISK original algorithm was conceived to operate on specific instances (i.e. pre-separated portions of the text containing the information to be extracted) while all the FE components developed inside the CROSSMARC project operate on the entire web pages. Heuristics that rely on semantic information provided by previous modules have been designed to limit the search space of induction [5].

c) Laplacian Expected Error versus Precision: rule appliance strategy. The Laplacian Expected Error Rate (i.e. $(e+1)/(n+1)$, where e is the number of wrong extractions and n is the overall amount of extractions made), originally adopted by Soderland for evaluating rules, has been preserved as the performance measure for evaluation of the temporary rules created during rule expansion, as it expresses a good trade-off between rule precision and recall while pure Precision is stored as an attribute for every rule, as it is necessary to establish which rules take precedence when they are applied outside the training phase.

5.1.2 Evaluation of Italian Fact Extractor component

At present time, CROSSMARC project is still to be concluded, but we made specific evaluation of FE components in the 4 different languages; in table 1 evaluation results for the domain covering laptop computer offers reports precision and recall statistics for all of the considered characteristics. The table below exposes black-box evaluation of the FE components, assuming optimal input from the previous processing steps (NERC and Product Demarcation);

Table 1: Evaluation results for the Laptop Computers Offers Domain on 4 different languages

FEATURE	ENGLISH		FRENCH		GREEK		ITALIAN	
	PR	RC	PR	RC	PR	RC	PR	RC
MANUFACTURER	0.89	1	0.99	1	1	1	1	0.99
PROCESSOR.	0.99	1	1	1	1	1	0.99	1
OP. SYSTEM	0.78	0.98	0.82	0.94	0.92	0.98	0.78	0.99
PROC.SPEED	0.86	0.99	0.95	0.98	0.85	1	0.95	0.98
PRICE	0.99	1	1	1	1	1	1	1
HD CAPACITY	0.99	0.94	0.94	0.80	0.96	0.96	1	0.88
RAM CAPACITY	0.82	0.97	0.95	0.94	0.90	0.80	0.96	0.89
SCREEN SIZE	0.85	0.98	0.70	0.99	0.95	0.98	0.92	0.99
MODEL NAME	0.99	1	1	0.99	1	1	0.99	1
BATTERY TYPE	1	0.86	0.97	0.63	0.97	0.76	1	0.5
SCREEN TYPE	0.82	0.98	0.81	0.96	0.99	1	0.86	0.99
WEIGHT	0.98	1	0.96	1	1	1	0.92	1
AVERAGE VALUES	**0.91**	**0,97**	**0.93**	**0.94**	**0.96**	**0.96**	**0.95**	**0.90**

6. The Knowledge model of CROSSMARC

All of the components described so far, are driven by a community of agents sharing common informational resources and semantics defined at different levels (i.e. lexical, ontological and task oriented). An ontological architecture has been developed upon this definition and has thus been organized around three different layers:

- a *meta-conceptual layer*, which represents the common semantics that will be used by the different components of the system in their reasoning activities
- a *conceptual layer* where relevant concepts of each domain are represented, and
- an *instances layer* where language dependent realizations of such concepts are organized.

The current ontology structure is maintained through a CROSSMARC application based on the APIs of Protégé 2000 [4], an ontology engineering environment that supports ontology development and maintenance. Protégé-2000 adopts a frame-based knowledge model, based on classes, slots, facets, and axioms.

The meta-conceptual layer of CROSSMARC defines how linguistic processors will work on the ontology, enforcing a semantic agreement by characterizing the ontological content according to the adopted knowledge model.

The Protégé metaclasses hierarchy has been extended introducing a few metaclasses. These are used in the Conceptual level to assign computational semantics to elements of the domain ontology. Basically the metaclass extension provides a further typization to concepts, adding a few constraints for formulating IE templates.

The instance layer represents both domain specific individuals and lexicalizations of these individuals into the adopted languages. It instantiates classes in the domain ontology; these instances fill the values for attributes of the domain templates.

7. Conclusions

The difficulty in building complex adaptive systems is represented by an unavoidable trade-off between how much experience and task-oriented skill must be put inside the system on one side, and how it must satisfy a certain degree of generality and the required openness versus possible extensions.

CROSSMARC aims to fulfil its requirements through a solid Knowledge Model provided with the necessary level of abstraction, which constitutes the main fabric through all of the agents that control its components can communicate, share information and cooperate to reach their tasks.

The neat separation between the Knowledge Model (the Meta-Conceptual Layer), the Domain Model (the Conceptual Layer) and Domain Instances and Languages (Values in the Ontology) permits easy plug-ability of different system resources and processors.

Following these premises, two techniques coming from two completely different approaches to IE have been integrated: Ontology and Language Driven Named Entity Recognition and Classification and Wrapper Induction Based Fact Extraction.

This way, the system takes the benefits of both the approaches: from a maintenance cost point of view, it is freed from the need for technical experts for customisation versus new domains and languages, leaving to knowledge engineers and domain experts the task of creating/updating ontologies and annotating other sites, at the same time, this combination permits to add the expressive power of Concept Recognition to Wrapper Induction processors, whose extracted elements would, in other case, remain meaningless strings.

10. References

[1] I. Muslea, S. Minton C. Knoblock "STALKER: Learning extraction rules for semistructured Web-based information sources". *AAAI-98*. Madison. Wisconsin.

[2] Sonderland S. "Learning Information Extraction Rules for semi-structured and free text." *Machine learning*. Volume 34 (1/3), 1999, pp. 233-272.

[3] Freitag D., Kushmerick N., "Boosted Wrapper Induction". In the Proceedings of the *7th National Conference on AI*, Austin, Texas, 2000.

[4] N. F. Noy, R. W. Fergerson, & M. A. Musen. "The knowledge model of Protege-2000: Combining interoperability and flexibility". *2th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000)*, Juan-les-Pins, France,2000.

[5] M.T. Pazienza, A. Stellato, M. Vindigni, "Combining Ontological Knowledge and Wrapper Induction techniques into an e-retail System", *ATEM2003 Workshop on Adaptive Text Extraction and Mining* 22 Sept. 2003 Cavtat-Dubrovnik, Croatia.