

Mining linguistic information into an e-retail system

M. T. Pazienza and M. Vindigni

DISP – University of Rome “Tor Vergata”, Italy

Abstract

Building more adaptive SW applications is a crucial issue to scale up Information Technology to the Web, where information is organized following different underlying knowledge and/or presentation models. To efficiently manage heterogeneous information sources agents must be able to cooperate, share their knowledge, and agree upon appropriate terminology to be used during interaction. We describe here an e-retail product comparison agent system aiming to supply users with synthesis information on product fitting at best their inquires. We will focus on the Named Entity Recognition and Classification (NERC) component that is very helpful in identifying relevant characteristics in a (multilingual) product description.

1 Introduction

The continuous growth of the Web accesses and e-commerce transactions are producing a new generation of sites: e-retail portals, willing to help end-users in choosing accordingly to their needs among different products, presented in an uniform way to make easier their comparison. A number of commercial agent-based systems already exist that help Internet shoppers in deciding what and where to buy goods. Most of these agents extract relevant data from on-line product descriptions, summarizing and presenting results in a synthetic form to the final user. They don't use a 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. Moreover, they often assume pages to be expressed in uniform and monolingual manner (usually English), this being unsuitable for a multi-lingual society such as the European one. This motivates spreading approaches and techniques to cover

areas of structured, textual, multimedia, web distributed data mining as well as data pre-processing, feature selection, feature transformation, man-machine interaction. In fact extracting semi structured data from e-retail sites (and in general from the Web) is a complex task, as information on the Web is usually organized for presentation purposes more than for automatic extraction systems. Images and texts, as a whole, contain the relevant information, organized in a fashion suitable to catch the human attention more than to be perceived as a rigorous and intelligible structure. The extraction task becomes even harder in our multi-lingual society, as web pages are typically written in different languages; moreover, since technology evolves, new product types are likely to appear in the market, requiring an expensive effort to customize existing systems/resources to new unforeseen scenarios.

We will describe here our contribution in building CROSSMARC, an e-retail product comparison agent system (currently under development as part of an EU-funded project), aiming to supply users with synthesis information on product fitting at best their inquires. The two main CROSSMARC goals are both in developing commercial-strength technologies based by language processing methodologies for information extraction from web pages and in providing automated techniques for an easy customization (extension of the system to new product domains and languages). CROSSMARC technology currently operates in 4 languages (English, Greek, French, Italian) and is being applied to two different product domains: computer goods and job offers. The first, covering laptop offerings, involves product descriptions written in a semi-structured way, in a technical and poorly structured language, while the second, IT job offers, contains wide free text descriptions broader in coverage. The two domains have been chosen to be as dissimilar as possible for presentation style, contents, use of tables and layout aspects to evaluate the system in large.

In the following an overall description of system architecture will be provided. To stress the role played by language processing techniques it will be deeply described the Named Entity Recognition and Classification (NERC) component: in fact recognition of domain specific named entities is very helpful in identifying relevant characteristics in a (multilingual) product description.

2 CROSSMARC Architecture

The overall CROSSMARC architecture implements a 3-tier model : (see fig. 1) the first tier being the presentation layer (user interface), the third the data processing layer (allocating several language processing components) and the middle the database structure where relevant information extracted is stored and maintained. Core components of the prototype system are:

- spider agents visiting the web and returning pages likely to contain product descriptions;
- high-quality Information Extraction (IE) components (for each language) which process product descriptions in XHTML pages, perform Named Entities Recognition and Classification and Fact Extraction and finally populate a database with information about vendors' offers;

- a user interface which processes the user's query, performs user modelling, accesses the database and supplies the user with product information.

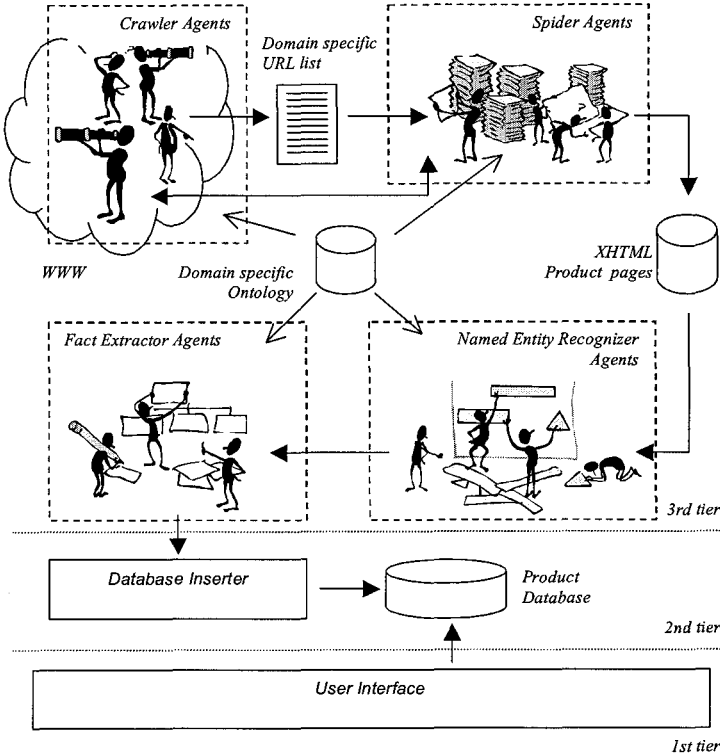


Figure 1: CROSSMARC architecture

A loosely coupled agent based architecture has been chosen both to easily extend the system over new languages, application domains, and/or services and due to the fact that foreseen activities are highly asynchronous (for example extracting information could be performed at fixed time, or when the overall load is low, while querying the previously extracted information could be a continuous task, executed any time). Individual monolingual components using XML to communicate each other have been plugged in: each partner contributes with his autonomous modules, exchanging information through a common vocabulary provided by a domain specific ontology.

In the following we will focus on the 3rd-tier as it relates to the complex processes of finding pages, recognizing features, extracting information and passing it to 2nd-tier for storing purposes.

What we need by first (to extract product offers from Web e-retail stores) is to locate and identify domain-specific retailers, then to navigate through their sites in order to recognize and collect Web pages describing user relevant products.

Once these Web pages have been collected, they could be processed by the information extraction components (i.e. named entity recognizer and fact extractor), that will extract relevant information to be stored in the product database in the 2nd tier.

2.1 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. laptop 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. Each visited page is evaluated and stored in a corpus while identified to be relevant, for further processing by the cross-lingual information extraction modules of CROSSMARC.

The result of this first process is a set of web pages (for each interesting web site) containing product descriptions, in form of HTML documents that must be converted to XML. This task is performed by passing from HTML to well-formed XHTML: for this purpose Tidy has been used; it performs the required translation, allowing for a first clean-up of messy structures (i.e. allowing to repair at some degree missing or mismatched tags, mixed up tags, and missing quotes around attribute values, and redundant presentational mark-up) often appearing in pages generated by WYSIWYG editors; then, XHTML pages are input to the linguistic processing components, that process them as (poorly specified) XML structures.

2.2 Multilingual NERC and Name Matching

The Multilingual NERC subsystem architecture [6b] is shown in Fig. 2 where the individual components are autonomous agents, which need not to be installed all on the same machine. Pages are routed to the appropriate agent, activated by the main CROSSMARC system, accordingly to the language they have been written in, thus allowing components running on other machines and possibly under different operating systems both to receive and send data. We distinguish between actions devoted to concept identification and those dealing with lexical entries. In fact, while all NERC components partition the task into a sequence of steps, each incrementally enriching the page with semantic tags referring to concepts in a common ontology, every NERC agent uses its own resources in the identification of entities and maintains its lexicon by adding synonym variants of the basic entries.

Each system differs in a few aspects, for instance the annotation method and used platform: English (E-NERC, see [6a] for details) and Italian (I-NERC, see [8]) are exclusively XML-based and their annotation method involves incremental transduction of the XML document using XML tools. On the other hand Greek (H-NERC, see [4]) uses Tipster-style annotations where tags are kept in a separate file along with pointers into positions in the document; moreover French (F-NERC, see [11]), H-NERC and I-NERC are Windows-based while E-NERC is Unix-based.

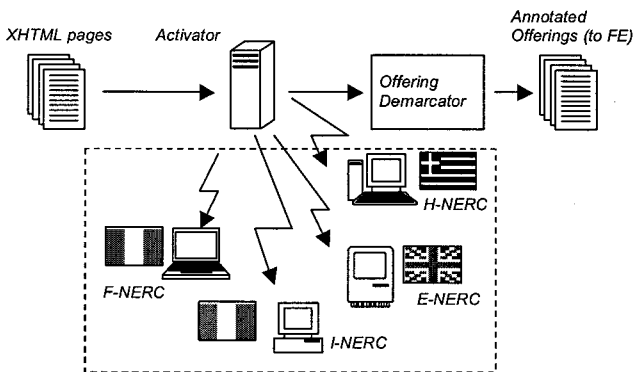


Figure 2: architecture of the Named Entity Recognizer Component

2.3 The Italian NERC

The I-NERC agent receives, from previous modules in the chain, the Web page as an XML structure containing description of one or more products from Italian retailers (see Figure 3), then processes and enriches it by adding markup tags for domain specific information (i.e., product name, manufacturer, cost, etc.) (see figure 4). 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. For terminological expressions (as *hard disk*, *CPU*, etc.) or frequently used proper names (i.e., *Compaq*, *Toshiba*, etc.), this is provided by background domain knowledge bases (i.e., gazetteers or terminological dictionaries), while for other categories (i.e., dates, measures, etc.), more declarative in nature, intentional descriptions (i.e. regular patterns) are provided.
- *External evidences* provided by the context in which the entity appears. For instance, the manufacturer's name often follows the word "*Notebook*", while the number following the processor's name is likely to be its speed; this kind of information is exploited by using contextual grammars, sets of rules to tag a text fragment by looking at previous/following words.

The relative impact these two different evidences have during the analysis heavily depends on the domain. In fact, domain knowledge represented by

gazetteers or terminological dictionaries, although highly accurate for name matching, has validity in time that depends on domain dynamics (new technologies, products and manufacturers, etc.). External evidences, although affected by uncertainty, provide a way to detect unforeseen values through context clues, and thus constitute a key for detecting also unknown names.



Figure 3: an Italian source offering fragment

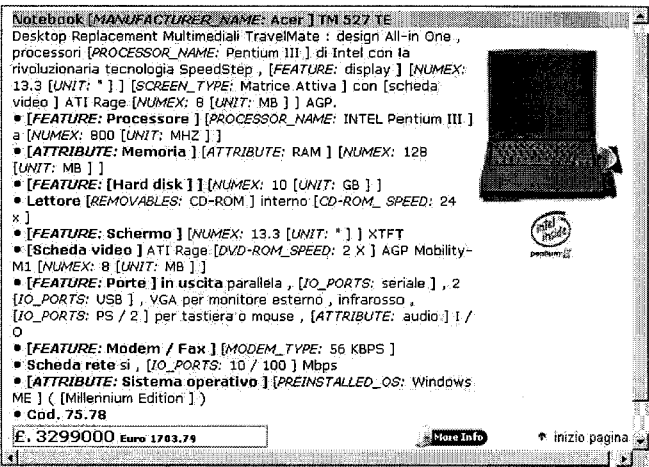


Figure 4: I-NERC annotations (pretty print) for the fragment of fig. 3

Contextual evidences can be grouped into two main classes:

- Structural evidences, provided by the document organization (tables, bullet lists, or, at a lesser extent, formatting properties as bold, italic, etc.).
- 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. This means that

XSLT transformations constitute control flow rules to navigate through the XML structure, selecting relevant sections to be analysed, and integrate the results in the overall document organization, while the linguistic processing is performed by dedicated components that operate their sub tasks and provide results autonomously.

2.3.1 I-NERC linguistic processing

The I-NERC agent is implemented as a Java application, using the TrAX API to control XSLT transformations (see Fig. 5). A sequence of linguistic processes are activated applying a pipeline of transformations on the page. Results of each step input the next one.

- 1) The **Normalizer** transformation provides a pre-processing at character-level of the source document, to deal with bad word separators, wrong punctuation and word-level misspelling or ungrammaticalities. For instance, in the laptop domain, it accounts for recognition and normalization of the decimal separator (very often both dot and comma are used, even in the same page), of the inch symbol (double vs. two single quotes), and special characters (as trademark and copyright symbols), that could later burden the analysis. Regular expressions are used to find such sequences and replace them with standardized representations.
- 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. In this phase, acronyms (as for instance MS for Microsoft, or IE for Internet Explorer, or HD for Hard Disk) are resolved, by creating complex XML elements that group the original tokens. In this way, following analysis could rely on a sort of “lemmatisation” and use a reduced number of rules to recognize complex expressions.
- 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. Here a first degree of semantic analysis is performed, as numeric entities are classified in broader categories (i.e. capacity, length, resolution, etc.) according to their units. Identification of related categories allows to define information for following Fact Extractor.
- 6) **Ontology-lookup** matches identified entities against the ontology and categorizes them accordingly. Ontology concept references are added to describe features, attributes or values the identified information belongs to.

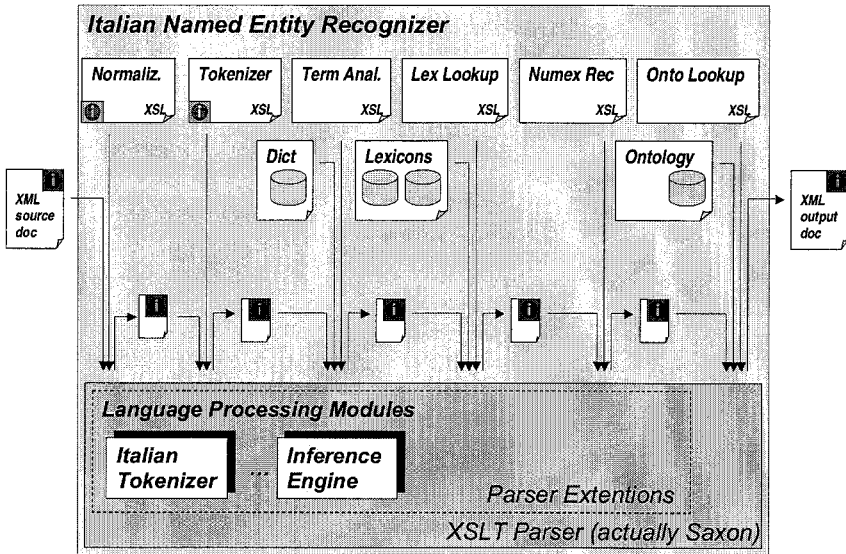


Figure 5: Internal organization of the Italian Named Entity Recognizer

Activities performed by I-NERC are not all strictly part of classical named entity recognition (for instance, normalisation of numerical and temporal expressions): several information is directly feed to the next Fact Extraction module as distinction between NERC and FE tasks depends on the methodology adopted. The strict linking of NERC and FE implies that decisions about when to perform a particular subtask are taken in the context of the combined NERC+FE components (agents). For instance, name matching could be postponed until the FE component, even though some part of the task is already performed at the NERC stage via use of the ontology-derived lexicons.

2.4 Multilingual and Multimedia Fact Extraction

The overall architecture of the Multilingual and Multimedia Fact Extraction (FE) system is analogous to NERC, being the input the XHTML web pages annotated by the NERC components. To better between balance between efficiency and restrictions of wrapper-based approaches for FE on one hand, and flexibility and complexity of language-based information extraction on the other, we planned to use wrapper induction techniques that exploit the results of the multilingual named-entity recognition and name matching component. With such an approach a wide variability in product names and feature expressions will be managed apart ability in capturing multimedia aspects of product descriptions (tables, hypertext links, etc.). The fact extraction involves the use of language-based techniques from information extraction in order to handle product descriptions written in free form.

3 Related work

The most common approach to extracting information from the web is wrapper induction ([7]). These systems mainly use delimiter-based approaches, assuming that processed texts convey information in a rigidly structured manner, with entities and features mentioned in a fixed order (e.g. product name always followed by price, then availability), and fixed strings or mark-up acting as delimiters. Techniques of this area are not applicable to product descriptions written in freer linguistic form, but they have proven to be very efficient with rigidly structured pages (for relevant experiments see [10]).

Within the field of Information Extraction and the NERC sub-task a variety of different approaches and a range of different domains and text types have been exploited. The Message Understanding Conference (MUC) competitions have been a highly visible forum for reporting IE and NERC results, see for example, MUC-7 ([2]). MUC systems were required to process newspaper texts, identify the parts relevant to a particular domain, and fill templates that contain slots for the events to be extracted and the entities involved. Information analysts design the template structure and fill manually the templates, which are then used in the evaluation. The current emphasis is on moving away from the rule-based approach, which relies on hand-crafted lexical resources and hand-crafted grammar rules, towards machine learning techniques in order to achieve swifter adaptation to new domains and text types. Predominant methodologies here foresee to employ a large amount of annotated corpus material to be used by the learning procedure(see, for example, [1], [5]). However, a few systems use machine learning over unannotated data ([3], [9], [12]).

CROSSMARC lies in between these approaches, aiming to combine wrapper induction techniques with language-based information extraction, with an emphasis to the rapid adaptation to new domains. CROSSMARC also operates in a cross-lingual environment: to match names referring to the same entities across different surface realisations in different languages, a domain ontology is used since this represents the common set of concepts, which play a role in the facts to be extracted.

Acknowledgements

CROSSMARC is a project of the Information Society Technologies (IST) Programme of the European Union (IST 2000-25366). CROSSMARC partners are NCSR "Demokritos" (Greece), University of Edinburgh (UK), Universita di Roma Tor Vergata (Italy), Informatique CDC (France), and VeltiNet (Greece).

References

- [1] M. E. Califf. 1998. Relational Learning Techniques for Natural Language Information Extraction Systems. Ph.D. thesis, University of Texas, Austin.

- [2] N. A. Chinchor. 1998. Overview of MUC-7/MET-2. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference held in Fairfax, Virginia*.
- [3] M. Collins and Y. Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.
- [4] D. Farmakiotou, V. Karkaletsis, G. Samaritakis, G. Petasis, and C. D. Spyropoulos. 2002. Named entity recognition in Greek web pages. In *Proceedings of the 2nd Panhellenic Conference on Artificial Intelligence, Thessaloniki, Greece*.
- [5] D. Freitag and A. McCallum. 1999. Information extraction with HMMs and shrinkage. In *Proceedings of the AAAI-99 Workshop on Machine Learning and Information Extraction*, pages 31–36.
- [6a] C. Grover, C. Matheson, A. Mikheev, and M. Moens. 2000. LT TTT— a flexible tokenisation tool. In *LREC 2000 — Proceedings of the Second International Conference on Language Resources and Evaluation, Athens*.
- [6b] C. Grover et al. 2002. Multilingual XML-Based Named Entity Recognition for E-Retail Domains. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC2002), Las Palmas*.
- [7] N. Kushmerick, D. Weld, and R. Doorenbos. 1997. Wrapper induction for information extraction. In *Proceedings of the 15th International Conference on Artificial Intelligence*, pages 729–735.
- [8] M. T. Pazienza and M. Vindigni. 2000. Identification and classification of Italian complex proper names. In *Proceedings of ACIDCA2000 International Conference, Monastir, Tunisia*.
- [9] E. Riloff and R. Jones. 1999. Learning dictionaries for information extraction by multi-level boot-strapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 1044–1049, Orlando.
- [10] S. Soderland. 1997. Learning to extract text-based information from the world wide web. In *Proceedings of 3rd International Conference in Knowledge Discovery and Data Mining (KDD-97)*, pages 251–254.
- [11] F. Wolinski, F. Vichot, and O. Gr'émont. 1998. Producing NLP-based on-line contentware. In *Natural Language Processing and Industrial Applications (NLP+IA)*, pages 253–259, Moncton, Canada.
- [12] R. Yangarber and R. Grishman. 2000. Machine learning of extraction patterns from un-annotated corpora. In *Proceedings of the Workshop on Machine Learning for Information Extraction, 14th European Conference on Artificial Intelligence (ECAI 2000)*.