Multi-lingual XML-Based Named Entity Recognition in Web Pages

Claire Grover, Donnla Nic Gearailt

School of Informatics, University of Edinburgh,
Edinburgh, UK
grover@cogsci.ed.ac.uk, donnlan@cogsci.ed.ac.uk

Vangelis Karkaletsis, Dimitra Farmakiotou

Institute for Informatics and Telecommunications,
National Centre for Scientific Research "Demokritos",
Athens, Greece

vangelis@iit.demokritos.gr, dfarmak@iit.demokritos.gr

Maria Teresa Pazienza, Michele Vindigni

D.I.S.P., Univerisita di Roma Tor Vegata,
Rome, Italy
pazienza@info.uniroma2.it, vindigni@info.uniroma2.it

Abstract

We describe the multi-lingual Named Entity Recognition and Classification (NERC) subpart of an information extraction system, which is currently under development as part of the EU-funded project CROSSMARC. The two main CROSSMARC goals are to develop commercial-strength technologies based on language processing methodologies for information extraction from web pages and to provide automated techniques for efficient customisation i.e. extension of the system to new product domains and languages. To achieve our goals we use XML as a common exchange format, we exploit a common ontology and the monolingual NERC components use a combination of rule-based and machine-learning techniques. It has been challenging to process web pages, which contain heavily structured data where text is intermingled with HTML and other code. Our evaluation results demonstrate the viability of our approach.

1. Introduction

The advent of e-commerce and the continuous growth of the WWW have given birth to a new generation of e-retail stores. The extraction of structured data from e-retail sites and in general from Web sites is a complex task. Most of the information on the Web today is in the form of HTML documents, which are designed for presentation purposes and not for automatic extraction systems. The extraction task becomes even harder in multi-lingual societies, where descriptions in web pages are typically written in different languages.

A number of systems have been developed to extract structured data from web pages. Such systems include a set of wrappers that extract the relevant information

from multiple Web sources and a mediator that presents the extracted information according to the users' requests. Most of these systems use delimiter-based approaches. Texts processed by them are assumed to 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. Though the techniques of delimiter-based approaches have proven to be very efficient with rigidly structured pages, they. are not applicable to product descriptions written in freer linguistic form. By contrast, CROSSMARC can operate on pages without a standardised format, as well as on pages from sites that have not been represented in the training corpus.

In this paper, we describe the multi-lingual Named Entity Recognition and Classification (NERC) component of an information extraction system, which is currently under development as part of the EU-funded project, CROSSMARC[1]. CROSSMARC aims to combine delimiter-based approaches (wrapper induction techniques) with language-based information extraction, giving emphasis to rapid adaptation to new domains. CROSSMARC also operates in a cross-lingual setting exploiting the domain ontology and the corresponding language-specific lexica. The core components of the CROSSMARC prototype system are:

- Web page collection tools which identify domain-specific Web sites (focused crawling) and navigate through them in order to identify Web pages of interest (domain-specific spidering);

[1] CROSSMARC (IST 2000 – 25366) is an R&D project on cross-lingual information extraction applied in e-retail product comparison, funded partially by the EC. CROSSMARC partners include NCSR "Demokritos" (coordinator), University of Edinburgh (UK), University of Roma Tor Vergata (Italy), Informatique CDC (France), VeltiNet (Greece), Lingway (France).

- a high-quality Information Extraction (IE) component for several languages which locates product descriptions in XHTML pages and performs NERC and fact extraction so as to populate a database with information about vendors' offers;
- a user interface which processes the user's query, performs user modelling, accesses the databases and presents product information back to the user.

Although NERC is a familiar task to the information extraction (IE) research community, there are novel aspects arising from our application area which present significant and interesting challenges both for implementation and for evaluation.

Section 2 reviews some related work. Section 3 describes the architecture of the multi-lingual NERC system and how this is integrated into the CROSSMARC prototype system. Section 4 discusses CROSSMARC extensibility to new languages and domains: we present the four monolingual NERC components which make up the CROSSMARC multi-lingual NERC system and we describe the process of adding new domains. Section 5 presents the CROSSMARC evaluation methodology and evaluation results for the first CROSSMARC domain. Section 6 concludes by summarizing our work and presenting our future plans.

2. Related Work

Within the field of Information Extraction (IE) and the NERC sub-task there are a variety of different approaches and a range of different domains and text types, which are processed. The Message Understanding Conference (MUC)[2] competitions have been a highly visible forum for reporting IE and NERC results, see for example MUC-7 ([4]). The systems participating in MUCs are required to process newspaper

[2] The most recent MUC results are available at

http://www.itl.nist.gov/iad/894.02/related_projects/muc/proceedings/muc_7_proceedings/overview.html

texts, identify the parts of a text that are 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 the templates manually, which are then used in the evaluation. NERC is the evaluation task for which the best results have been achieved, proving that this technology is mature. The entities to be recognised for the NERC task in MUCs are ENAMEX (people, organisations and locations), TIMEX (dates and times) and NUMEX (money and percentages). Approaches to the MUC NERC task range from the purely rule-based to the purely machine-learning based, with hybrid systems combining rules and machine learning in between. Overall, combined precision and recall for the best MUC NERC systems is around 93%, which is nearly comparable with human performance. However, the systems competing are likely to have been highly tuned to the domain and would not port easily to new domains or new text types.

In the wider field of NERC, 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. The predominant method has been to create a large amount of annotated corpus material to be used by the learning procedure (see, for example, [3], [8], [21]). The alternative to this is to use machine learning over un-annotated data ([5], [19], [25], [26).

Recently, there has been increasing interest in the web as an application area for IE technology, as is the case for CROSSMARC. Web pages differ from raw text in terms of content and presentation style. Apart from raw text, they also contain links, images and buttons. Statistical corpus analysis has shown that hypertext forms a distinct genre of linguistic expression following separate grammar, paragraph and

sentence formation rules and conventions. Such differences can affect the performance of standard NLP techniques when transferred to hypertext ([1], [20]).

An informal comparison of a corpus of Web pages to flat texts of the same domain (descriptions of laptops from computer magazines) in the context of CROSSMARC showed the following:

- Hypertext paragraphs and sentences are usually much shorter than the ones frequently encountered in free text.
- Itemized lists and tabular format are used more frequently in hypertext than free text.
- On-line laptop descriptions require more domain knowledge on the part of the reader than flat text descriptions.
- A vast number of on-line descriptions of computer goods present the reader with phrase fragments and numeric expressions without their measurement units e.g. “P3 800 256 14 TFT”, whereas flat text descriptions contain complete sentences and phrases like “a Pentium III processor with 256 MB of RAM” that facilitate text understanding. Full phrases contain contextual information for the classification of NEs, whereas phrase fragments found in web pages require more knowledge of the writing conventions (e.g. a number following the name of a processor is the processor’s speed) and names that are easier to recognize must be used as the context for other possible names or expressions of interest.
- Unlike flat text that is processed word by word, the processing of hypertext documents is conducted in a web page source. A web page source is typically comprised of HTML tags intermingled with free text and JavaScript (or other) code.

The HTML parts of a page contain layout information that can be crucial for NERC and fact extraction. For example, knowing that two cells of a table are adjacent in the same row may help a system decide that the contents of the second cell are names of operating systems or software packages if the key phrase “Pre-installed Software” comprises the contents of the first cell. Thus, the incorporation of layout information is important in the adaptation of a NERC system to the genre of hypertext. The fact that HTML documents are frequently far from well-formed imposes greater difficulty in their processing and makes necessary their pre-processing into well-formed (XHTML) pages.

HTML tags have been used as an exclusive means for name recognition and identification in the creation of wrappers (for a formal description of some types of wrappers see [11]). The most common approach to extracting information from the web is the training of wrappers using wrapper induction techniques ([10], [15]). The drawback to this method is that it is web-site specific and, moreover, it can only be successfully applied to pages that have a standardised format and not pages that present a more irregular format.

CROSSMARC attempts to balance the use of HTML layout information with the use of linguistic information in order to enable NERC in both rigidly and less rigidly formatted types of pages. For this reason considerable effort has been placed on the selection of the HTML tags that are likely to convey important layout information and to the coding of a non-linear text format (e.g. tabular format) to a linear representation that enables the use of linguistic processing.

3. The Multi-lingual NERC System

The Multi-lingual NERC system is a module of the integrated CROSSMARC prototype shown in Figure 1. The whole application is divided into three tiers. The

first tier is the presentation layer, the third tier is the data-producing layer (web pages collection, named entity recognition and fact extraction) and the middle tier is the database structure that both these layers use to communicate.

As Figure 1 shows, interesting (domain-specific) 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 if it describes a product, it is stored in a corpus of web pages (for each e-retailer site). These pages are HTML documents that, according to CROSSMARC specifications, must be converted to XML. We are currently using a tool named meta-tidy, which is actually a wrapper to the Tidy[3] program in order to convert to well-formed HTML (XHTML). Therefore, the input to NERC consists of web pages collected by the collection tool, which have been converted to XHTML.

Figure 2 shows the overall architecture of the Multi-lingual NERC system. The NERC system is comprised of each of the individual monolingual components, and pages are routed according to the language they are written in. The architecture of the integrated multi-lingual NERC system is a distributed one where the individual components are autonomous processors, which need not all be installed on the same machine. The main CROSSMARC system is able to call each monolingual NERC component using a simple client-server mechanism, thereby allowing components running on other machines and possibly under different operating systems to receive and send data.

[3] http://www.w3.org/People/Raggett/tidy/

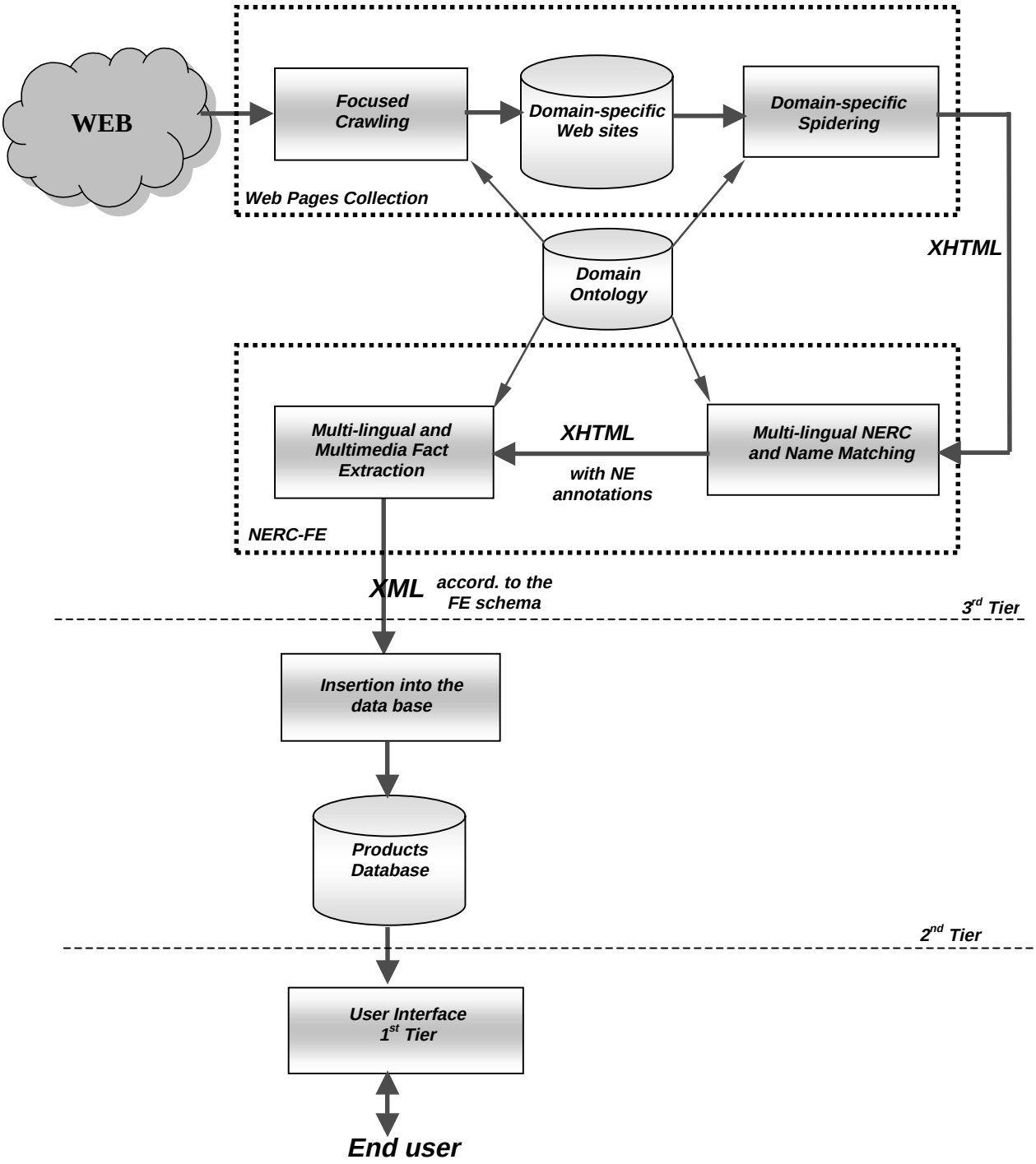

Figure 1 The 3-tier architecture

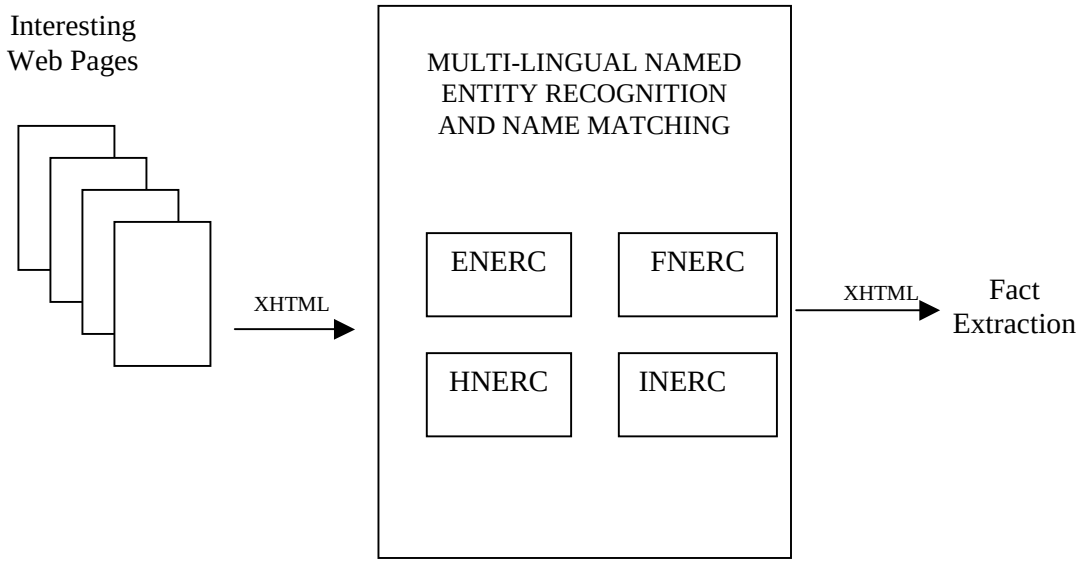

Figure 2 Architecture of the Named Entity Recognition Module

The monolingual NERC components are described in Section 4 (ENERC=English NERC, FNERC=French NERC, HNERC=Hellenic NERC and INERC=Italian NERC). The architectures of these systems are very similar in that they all partition the task into a sequence of steps which incrementally add information to the mark-up. In all cases the text is first segmented into words tokens and some kind of word level analysis is achieved, be it lemmatisation or part-of-speech tagging. Building on the word-level analysis, NERC rules are applied and these are typically composed of regular expression matching combined with lexicon or gazetteer look-up plus use of surrounding context. For each language there is a lexicon which is derived from the domain ontology and which contains links into the ontology in the form of ids. Each NERC system uses the appropriate lexicon in the identification of entities and maintains its lexicon by adding synonym variants of the basic entries.

The systems differ in certain respects, the most obvious differences being in annotation methods and platforms. With regard to annotation methods, ENERC and INERC are exclusively XML-based and their annotation method involves incremental transduction of the XML document using XML tools: in the case of INERC, these are standard XSLT tools and, in the case of ENERC, these are in-house, though publicly-available, tools. HNERC, on the other hand, uses Tipster[4]-style annotations where the tags are kept in a separate file along with pointers into positions in the document, though this annotation is easily converted to XML. FNERC is implemented as a Perl program which incrementally performs regular expression matching and which is able to output annotations either in the Tipster style or as XML mark-up. As regards platforms, HNERC and INERC are Windows-based, ENERC is Unix-based and FNERC runs on both platforms. A difference in platforms is not problematic since the distributed prototype performs remote invocation of the NERC systems running on different machines.

The NERC modules produce two types of output. The first is an XML document conforming to the specifications of the NERC DTD which is useful for evaluation purposes. The second is the XHTML page enriched with named entity annotations which is used for feeding the NERC-based demarcation tool. This tool is responsible for the discovery of the part(s) of a web page, which constitute product offerings, i.e. information about a specific offer of a specific product at a specific price.

The CROSSMARC partners are currently developing version 1 of Fact Extraction (FE) and are experimenting with the use of wrapper induction techniques. A typical Wrapper induction system generates delimiter-based extraction patterns that

[4] http://www.itl.nist.gov/iaui/894.02/related_projects/tipster/

do not use linguistic constraints. We are examining the combination of these delimiter-based extraction patterns with linguistic-based extraction patterns, exploiting the results of the CROSSMARC NERC component. That is, the wrapper induction system is trained using XHTML pages annotated with named entity annotations. The resulting trained wrapper is then used to extract information from XHTML pages already processed by CROSSMARC NERC components. The output of the FE system is an XML document according to the FE schema specified for the domain. This schema determines the facts to be extracted and links them with the NERC DTD and the ontology schema. The FE output feeds the Inserter module of the 2nd tier of CROSSMARC architecture (see Figure 1) which creates SQL INSERT statements for the product descriptions contained in the output. This is necessary in order to insert the extacted information from the product descriptions into the product database

4. Extensibility

The main goal in designing the system is that it should be rapidly extensible both to new languages and to new domains. In order to support both of these requirements we use XML as a common exchange format. For each product domain, we define an ontology of the product type using XML schemas, and this ontology is used to shape both the product database and the XML DTDs and schemas which define the common input and output of the NERC and fact extraction modules.

In a multi-lingual system like CROSSMARC, there inevitably arise questions of localization. Some localization aspects of our task, for example the fact that we need different character sets (Greek alphabet, accented characters), follow straightforwardly from XML's character encoding capabilities. Other localization issues require special strategies. In CROSSMARC we need to ensure that we can

match names that refer to the same entities across different surface realisations in the different languages. For example, the following all describe the same battery type: *Lithium Ion, Ions Lithium, Ioni di litio, Ιόντων* **υ** . We use the domain ontology as the means to match names across languages since this represents the common set of concepts which play a role in the facts we aim to extract. The ontology is represented as an XML document with constraints on it coded by means of the schema which defines it. Each node in the ontology has an attribute containing a distinct id number and one of the tasks of combined NERC and fact extraction is to link named entities with the relevant ontology node by encoding the id as an attribute value in the XML mark-up. This serves not only to match different surface realisations of the same concept across languages, but also to match them within the same language. Thus *Mobile Intel Pentium III* and *P3* will both be linked to the same concept in the ontology. The CROSSMARC ontology is maintained using the Protégé-2000 knowledge base editing environment ([14])

4.1. Multi-linguality

The prototype currently includes English, French, Greek and Italian but it must be possible to integrate IE components for other languages with a minimum of difficulty. For this reason, the individual monolingual NERC modules which make up the multi-lingual NERC component are only loosely coupled together and the only constraints that are placed on them concern input and output formats. Each monolingual NERC module takes an XHTML page as input and returns the same page augmented with XML annotations marking the named entities which it has found in the page. Thus each group must contribute a module which is capable of handling XML input and output but in other respects the systems can differ quite significantly. The four current

monolingual NERC modules all have different compositions and rely on different background technologies and data models. A brief description of each follows.

English NERC (ENERC)

ENERC is exclusively XML-based and the annotation process involves incremental transduction of the XHTML page. The page is passed through a pipeline which is composed of calls to a variety of XML-based tools from the LT TTT and LT XML toolsets ([9], [22]) as well as the *xmlperl* program ([12]). The system architecture is similar to the hybrid system which was the LTG's contribution to the MUC-7 competition ([13]): early processing stages use hand-crafted rules for 'sure-fire' named entities while a statistical classifier is used in the later stages to resolve more difficult potential named entities. The processing steps involve word-level tokenisation, part-of-speech tagging, the re-use of existing rules for date and time entities and the use of specialised rules sets for identifying domain specific entities such as laptop manufacturers, processor names, processor speeds etc. The pipeline has been made sensitive to the fact it is web pages that are to be processed and it targets

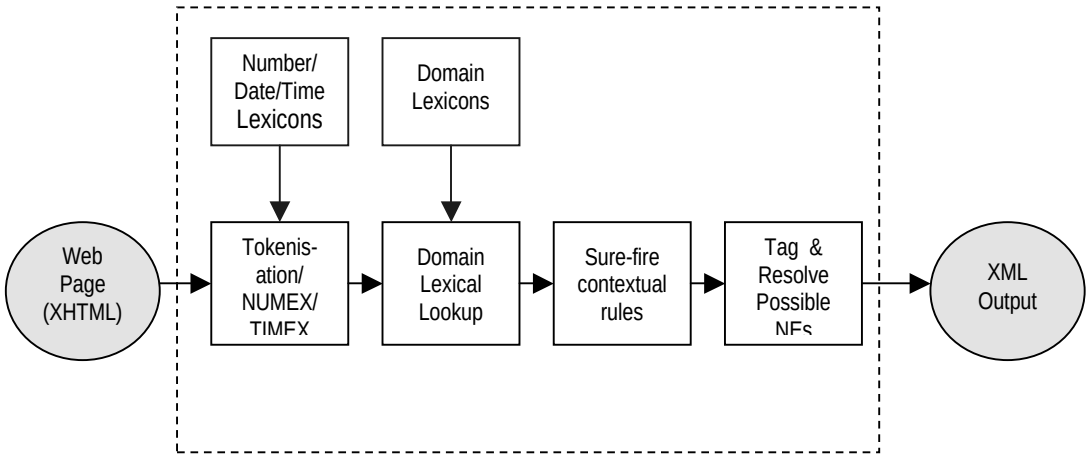

Figure 3. ENERC System Architecture

specific rule sets at specific sub-parts of the XHTML tree structure. The architecture of ENERC is shown in Figure 3.

French NERC (FNERC)

FNERC builds on previous work and reuses some techniques of the TalLab platform, which is I-CDC's Perl multi-agent framework for the development of NLP applications ([23], [24]). FNERC has been implemented as a standalone module in order to ease its integration into the CROSSMARC multi-lingual platform. It takes as input XHTML pages produced by Tidy from HTML pages. Due to the wide variety of different web sites and the highly-structured nature of the XHTML, an initial phase which is sensitive to the different types of XHTML elements (tables etc.) performs normalization of certain parts of the page in order to allow the next stages to proceed. This normalization is domain independent. At the next stage, the module performs named entity identification for three kinds of entities: NE, NUMEX, TIMEX. Some entities are domain independent such as dates, times and prices while others are domain specific such as laptop manufacturers, processor speeds, capacities, etc. Identification and classification of the named entities is performed by a wide set of regular expressions derived from the XML laptop ontology and language dependent lexicons. The final output is an XHTML file which is input to a demarcation tool which identifies individual product offers.

Hellenic NERC (HNERC)

HNERC has been developed using the Ellogon text engineering platform ([18], [7]). XHTML Web pages are converted into collections of Ellogon documents and the HNERC system applies a set of modules that add Tipster-style annotations to the Ellogon document, produce an XML document with the system output conforming to

the NERC DTD and add named entity mark-up to the original XHTML input page. HNERC has been adapted to web pages and the laptop domain from the MITOS-NERC system ([7]) which was designed for NERC in Financial News Texts. The Ellogon modules comprising HNERC perform Lexical Preprocessing, Gazetteer Lookup, Name identification and classification for NE, NUMEX and TIMEX (see Figure 4).

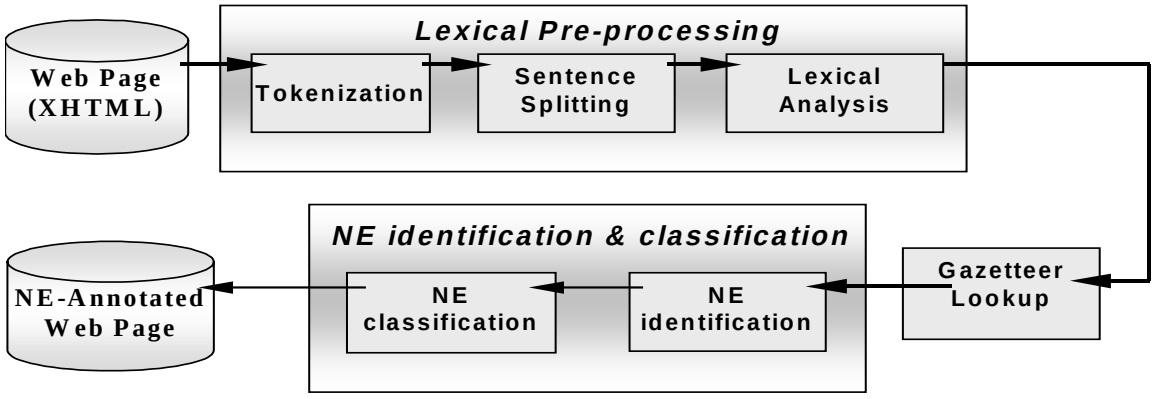

Figure 4. HNERC System Architecture

Lexical Preprocessing includes Tokenization, Zoning, Part of Speech Tagging, Lemmatization and Sentence Splitting. The modules performing tokenization, zoning and sentence splitting have undergone substantial adaptations. The set of tokens which the Tokenizer module originally handled has been expanded to include token types found in JavaScript (or other) code, HTML and HTML entities. Zoning has also been extended to include identification of text zones such as paragraphs, titles, images, tables, and table cells, rows and columns. Sentence splitting has been adapted in order to take into account structured information layout, e.g. tables, where fragmented sentences exist. The Gazetteer Lookup lists have also been updated to accommodate the needs of the new domain. Based on the results of the Lexical

Preprocessing and Gazetteer Lookup stages, the NERC parser constructs an internal representation of the processed document and performs Identification and Classification of names and expressions by applying pattern grammars to the internal representation. Identification and Classification of unambiguous names and expressions is performed first, whereas ambiguous names and expressions are identified and classified at subsequent stages. At a later stage bootstrapping is employed for the classification of possible names using classified names as contextual cues.

Italian NERC (INERC)

The INERC component ([16]) is implemented as a sequence of processing stages driven by XSLT transformations over the XML input structure, by using an XSLT parser (currently Saxon) with a number of language processing modules plugged in as extensions. XSLT transformations provide the control mechanism to apply linguistic processors to specific document sections; they drive the analysis through the XML structure and select relevant sections to be analysed, while linguistic processing is usually performed by specific components which return their results to be properly inserted. The linguistic processors which are currently utilized in the INERC component include a word tokenizer, a terminology analyser (which performs resolution of terminological expressions as well as acronyms), a lexical analyser, a module that matches identified Named Entities against the domain ontology to classify their roles, and dedicated parsers for NUMEX and TIMEX expressions (see Figure 5). Other components, such as an Italian part-of-speech tagger, a chunker and a shallow parser ([2]) can also be inserted in the processing chain, though they have not been required in this first domain since the laptop corpus pages are linguistically

impoverished. Almost all of the INERC system is implemented as a Java application, using the TrAX API to control XSLT transformations.

The overall architecture of the INERC component is shown in Figure 5; more details can be found in [17].

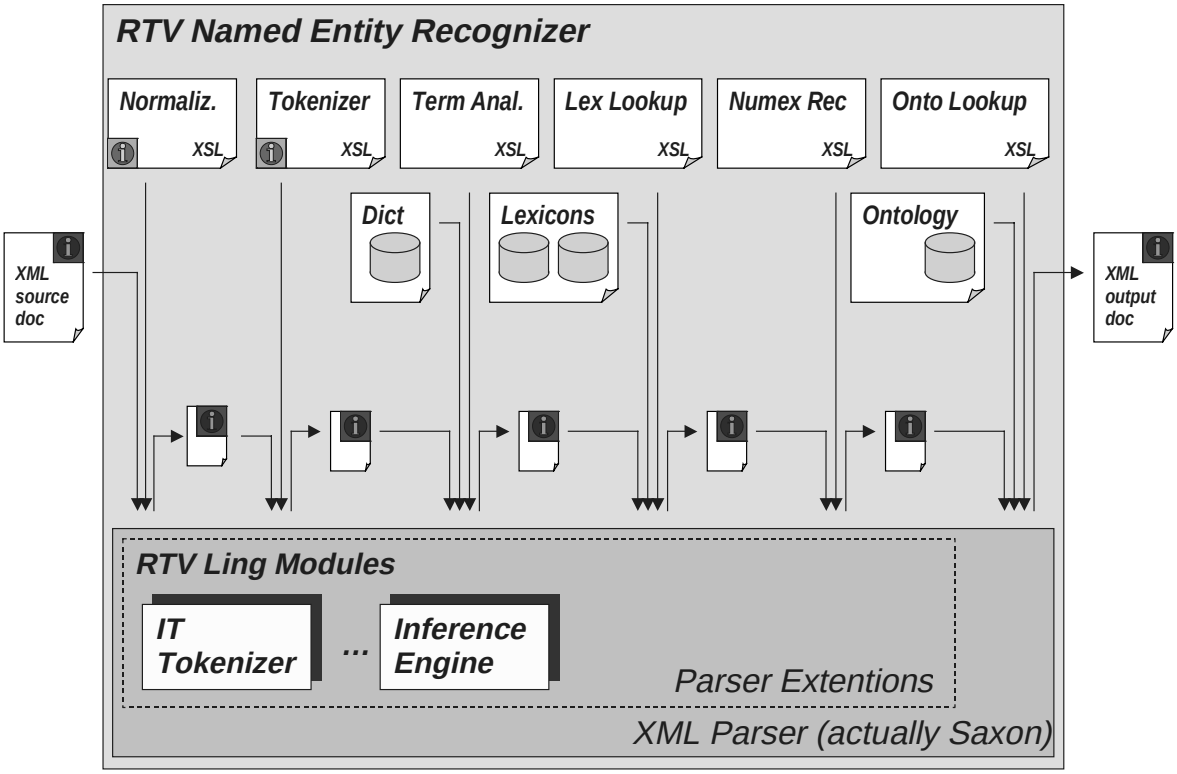

Figure 5. INERC System Architecture

4.2. Adding New Domains

At this stage in the project we have produced a first version of NERC for the first product domain (laptop computers) and we have just started work on the second domain (job adverts on companies' web pages). The two domains have been chosen to be as dissimilar to one another as possible in terms of presentation style, amount of discursive text, use of tables and other layout devices etc. This experience of significantly different styles of web page will contribute to our aim of making it as swift and easy as possible to add new product domains. As can be seen from Figure 1,

each domain is characterised by a domain ontology in order that a fixed amount of information can be extracted from the web pages and inserted into the product database. This provides key facts about the products to be used as the basis of comparison and in presenting information to the user. The combined use of the ontology and database determines the templates which must be filled by the fact extraction module, and this in turn largely determines which named entities are to be recognised by the NERC component. The minimum set of entities which must be recognised by the NERC components for the laptop domain and for the job adverts domain are shown in Tables 1 and 2 respectively.

Entity	Sub-types
NE	MANUF, MODEL, PROCESSOR, SOFT_OS
TIMEX	TIME, DATE, DURATION
NUMEX	LENGTH, WEIGHT, SPEED, CAPACITY, RESOLUTION, MONEY, PERCENT

Table 1 Minimum Set of NERC Entities for the 1st domain

Entity	Sub-types
NE	LOCATION, ORGANIZATION, JOB_TITLE, EDU_TITLE, LANGUAGE
TIMEX	DURATION

Table 2 Minimum Set of NERC Entities for the 2nd domain

Some of these labels apply to strings, which can be used in more than one way. For example a string recognised as a <NUMEX TYPE=´SPEED'> may describe processor speed, CD-Rom speed, DVD-Rom speed or modem speed, depending on its context. The fact extraction module that follows NERC is responsible for disambiguating these kinds of cases. We view the set of entities in Tables 1 and 2 as a

minimum set of requirements on the individual NERC components. They represent the most salient characteristics of the domains and they are the entities that occur most consistently and most frequently in templates for all the languages. By ensuring that these entities are reliably recognised by the NERC component we lay an appropriate foundation to meet the core needs of the subsequent fact extraction component. Defining this minimal set also helps with evaluation of NERC performance by providing a fixed set of entities as the basis for comparison. Section 5 provides details concerning evaluation. Since the set of entities in Tables 1 and 2 is a minimum set, it follows that the individual NERC components are free to compute more information. In practice, the boundary between monolingual NERC and subsequent fact extraction may be quite fluid and NERC developers may want to recognise other entities which play a role in fact extraction but which can be computed during NERC. For example, the fact extraction schema makes provision for non-core facts such as information about the modem type or the battery type or the period for which a warranty is valid, and it is open to the NERC developers to recognise the entities involved if it is possible at this early stage.

5. Evaluation

For the creation of the necessary training and testing corpora, the “Web Annotator” tool was developed. This is designed to assist human annotators in tagging the named entities found in the web pages. The tool takes as input the ontology and the NERC DTD for the relevant domain (e.g. the laptop ontology and DTD) and converts the names of the entities into a clickable menu. The annotator uses the mouse to select a string in the document and then chooses an appropriate label from the menu. Annotations are stored as byte-offset files in the style of the Tipster architecture but can also be merged directly into the XHTML document as XML elements. A corpus

of web pages belonging in the 1st domain (laptops) was collected for each language and the Web Annotator was used to add annotations. Tags to identify the boundaries of each distinct product description in the page were also added. The four language corpora for the 1st domain that resulted from this phase differ in size and in content in some respects. Each group divided their corpus into training and testing material and details of the test corpora are summarized in Table 3.

Language	*No. Sites*	*No. Pages*	*Products*
English	13	23	151
French	7	22	87
Greek	17	84	109
Italian	9	15	90

Table 3: The Test Corpora for the 1st Domain

The testing corpora differ significantly in terms of the number of pages that have been included, though the number of pages does not directly relate to the number of product descriptions contained within them. Table 3 shows that there are clear differences in presentation style between the corpora, with Greek pages having a strong tendency towards one product per page and English and Italian pages tending to contain long lists of products on one page. Thus the English and Italian test corpora contain many fewer pages but more product descriptions. The French corpus lies somewhere in between. In evaluating the mono-lingual NERC systems we follow standard practice in the IE field of comparing system output against the hand-annotated gold-standard and measuring precision and recall for each category of named entity. Recall is a measure of how many entities from the gold-standard were marked up in the system output and precision is a measure of how many of the entities in the system output actually occur in the gold-standard. It is possible for a system to score well for recall (i.e. finding a high number of the entities which are marked up in

the gold-standard) while scoring badly for precision (i.e. marking up high numbers of entities which are not marked up in the gold-standard). Conversely, a system might achieve high precision (i.e. not finding entities which are not marked up in the gold-standard) but low recall (i.e. failing to mark up a large proportion of entities which are marked up in the gold-standard). Different applications may require a bias in favor of either recall or precision. The standard way to score a system's performance in general is to compute F-measure which averages across recall and precision. More precisely, F = 2*(recall*precision)/(recall+precision).

Each NERC developer performed evaluation of their system against their test corpus and calculated the relevant results for recall, precision and f-measure for each individual category of named entity. A glance at these results reveals similar performance levels for the systems, with F-measure usually occurring in the range 75%-85%. In Table 4 we average across all four monolingual systems to provide evaluation results for version 1 of the CROSSMARC NERC system as a whole.

		Precision	*Recall*	*F-measure*
NE	MANUF	0.77	0.89	0.83
	MODEL	0.87	0.59	0.70
	SOFT_OS	0.84	0.79	0.81
	PROCESSOR	0.92	0.95	0.93
NUMEX	SPEED	0.85	0.78	0.81
	CAPACITY	0.84	0.85	0.84
	LENGTH	0.87	0.80	0.83
	RESOLUTION	0.85	0.84	0.84
	MONEY	0.97	0.77	0.86
	WEIGHT	0.93	0.85	0.88
TIMEX	DATE	0.81	0.67	0.73
	DURATION	0.87	0.69	0.77
ALL		0.87	0.79	0.83

Table 4. Evaluation Results for the 1st Domain

From Table 4 it can be seen that F-measure for most categories falls in the range 75%-85%, with only MODEL falling below that range and PROCESSOR, MONEY and WEIGHT falling above it. The average recall across all categories is 79%, with the average precision at 87%. Thus as a whole our system tends towards gaining precision at the expense of recall. The overall F-measure indicates a system accuracy of around 83%, which compares very favorably with the best recorded MUC scores of around 93%. We find these scores very encouraging given that they are for version 1 of a system which deals with a very new domain and text type. Within the individual categories, the MODEL entity scored lowest. This is unsurprising since there are no specific models listed in the ontology or lexicons derived from the ontology. While other entities remain quite stable across product descriptions, model names and numbers tend to vary to such an extent that it was decided not to attempt to list models in the ontology. Model names, such as *Vaio* when associated with *Sony* and *Portege* when associated with *Toshiba*, are relatively easy to recognise but the alphanumeric strings that indicate model number (e.g. PCG-FX240K) are harder to recognise and to be sure of their boundaries. Since our evaluation method only recognises exact matches, any partial matches that may occur in MODELs, or indeed in other categories, are counted as failures. In this respect the evaluation measure makes a harsher judgment than strictly necessary: since NERC forms part of the IE processing chain, with fact extraction operating on its results, its main purpose is to provide a reliable source of information to bootstrap further reasoning. It is quite possible that in some cases partial matches would also be useful for the subsequent fact extraction task. We expect NERC performance to be improved in up-coming versions. We anticipate an increase in the size of the training and testing corpus for each language and this, combined with name matching and improved use of machine

learning techniques, as well as improvements to the tools used by the monolingual NERC components, will help us to reach this objective.

Apart from the evaluation of the monolingual NERC modules using recall, precision and F-measure, we also organized a preliminary evaluation of the CROSSMARC prototype system by users outside the CROSSMARC consortium. In the context of the Summer Convention on Information Extraction (SCIE-2002)[5], we organized an evaluation event for CROSSMARC where we had the chance to present CROSSMARC technologies and ask SCIE participants to evaluate some of these technologies (spidering of web pages, monolingual IE systems, user interface)[6]. The users were able to feed the monolingual IE systems (these are composed by the NERC and FE components) with web pages containing product descriptions (laptops) and evaluate the extracted information. The results were very satisfactory although we were at that time at an intermediate stage of development of the 1st version of CROSSMARC prototype. We are currently organizing the user evaluation of the 1st version which will involve a larger number of users based on the experiences from the first user evaluation event in the context of SCIE-2002.

6. Concluding Remarks

In this paper, we have described the CROSSMARC NERC system, which integrates, according to the project architecture, four named-entity recognisers (English, French, Hellenic, Italian) configured for the 1st domain of the project (laptops).

[5] http://scie02.info.uniroma2.it/

[6] http://www.iit.demokritos.gr/skel/crossmarc/external/crossmarc_event.htm

While the CROSSMARC partners all have a strong background in NERC in linguistic domains (e.g. the MUC task), we have found the move to working with web pages quite complex and challenging. Web pages differ from more standard text types in terms of both content and presentation style. These differences can affect the performance of standard NLP techniques. CROSSMARC partners have ported their NERC technologies from raw text to web pages taking into account the differences introduced by the new text genre. Our approach compares favorably with other methods of information extraction from Web pages, such as standard wrapper induction, because it is not site-specific and it can be used on pages with irregular formats which have not been seen before in the training material. Our multi-lingual system is rapidly extensible both to new languages and to new domains.

The evaluation results of the monolingual NERC systems reveal similar performance levels for the systems, with F-measure usually occurring in the range 75%-90%. We expect NERC performance to be improved in subsequent versions. The foreseen increase in the size of training and testing corpus per language, the incorporation of name matching and machine learning techniques as well as the improvement of the tools used so far by the monolingual NERC components will help us reach this objective.

References

[1] E. Amitay. 1997. Hypertext: The importance of being different. Master's thesis, Centre for Cognitive Science, University of Edinburgh, September.

[2] R. Basili, M. T. Pazienza, M. Vindigni, and F. M. Zanzotto. 1999. Adaptive parsing for time-constrained tasks. In Proceedings of the Associazione Italiana per l'Intelligenza Artificiale (AI*IA99), Bologna, Italy.

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

[4] 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, 29 April–1 May, 1998. http://www.muc.saic.com/ proceedings/muc_7_toc.html.

[5] 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.

[6] D. Farmakiotou, V. Karkaletsis, J. Koutsias, G. Sigletos, C. D. Spyropoulos, and P. Stamatopoulos. 2000. Rule-based named entity recognition for Greek financial texts. In Proceedings of the Workshop on Computational Lexicography and Multimedia Dictionaries (COMLEX 2000), pages 75–78, Patras, Greece.

[7] 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.

[8] D. Freitag and A. McCallum. 1999. Information extraction with HMMs and shrinkage. In Proceedings of the AAAI-99 Workshop on Machine Learning and Informa-tion Extraction, pages 31–36.

[9] 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, 31 May – 2 June 2000, Athens, pages 1147–1154. http://www.ltg. ed.ac.uk/papers/00tttlrec.pdf.

[10] 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.

[11] N. Kushmerick. 1997. Wrapper Induction for Information Extraction. Ph.D. thesis, University of Washington.

[12] D. McKelvie. 1999. XMLPERL 1.0.4. XML processing software. http://www.cogsci.ed.ac. uk/˜dmck/xmlperl.

[13] A. Mikheev, C. Grover, and M. Moens. 1998. Description of the LTG system used for MUC-7. In Seventh Message Understanding Conference (MUC–7): Proceedings of a Conference held in Fairfax, Virginia, 29 April–1 May, 1998. http://www.muc.saic.com/ proceedings/muc_7_toc.html.

[14] Noy N.F., Fergerson R.W., and Musen M.A. (2000) The knowledge model of Protege-2000: Combining interoperability and flexibility. Proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000), Juan-les-Pins, France.

[15] I. Muslea, S. Minton, and C. Knoblock. 1998. Stalker: Learning extraction rules for semistructured, web-based information sources. In Proceedings of AAAI-98 Work-shop on AI and Information Integration, Madison, Wisconsin.

[16] M. T. Pazienza and M. Vindigni. 2000. Identification and classification of Italian complex proper names. In Proceedings of ACIDCA2000 International Conference, Monastir, Tunisia, March.

[17] M. T. Pazienza, and M. Vindigni. 2002. Mining linguistic information into an e-retail system. Proceedings of the Conference DATA MINING 2002, Bologna, September 2002.

[18] G. Petasis, V. Karkaletsis, G. Paliouras, I. Androutsopoulos, and C. D. Spyropoulos. 2002. Ellogon: A new text engineering platform. In Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002), Las Palmas, Canary Islands, Spain.

[19] E. Riloff and R. Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In Proceedings of the Sixteenth National Conference on Artificial Intelligence, pages 1044–1049, Orlando, FL.

[20] 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.

[21] S .Soderland. 1999. Learning information extraction rules for semi-structured and free text. Machine Learning, 34:233–272.

[22] H. S. Thompson, R. Tobin, D. McKelvie, and C. Brew. 1997. LT XML. Software API and toolkit for XML processing. http://www.ltg.ed.ac.uk/ software/.

[23] F. Wolinski and F. Vichot. 2001. Des multi-agents pour developper des applications de contenu en ligne. Technique et Science Informatiques, 20(2):213–232.

[24] F. Wolinski, F. Vichot, and O. Gremont. 1998. Producing NLP-based on-line contentware. In Natural Language Processing and Industrial Applications (NLP+IA), pages 253–259, Moncton, Canada.

[25] 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).

[26] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. 2000. Automatic acquisition of domain knowledge for information extraction. In Proceedings of COLING 2000, Saarbruecken.