Multilingual XML-Based Named Entity Recognition for E-Retail Domains

Claire Grover[*], **Scott McDonald**[*], **Donnla Nic Gearailt**[*],
Vangelis Karkaletsis[†], **Dimitra Farmakiotou**[†], **Georgios Samaritakis**[†], **Georgios Petasis**[†],
Maria Teresa Pazienza[◇], **Michele Vindigni**[◇],
Frantz Vichot[‡], **Francis Wolinski**[‡]

[*]Language Technology Group, University of Edinburgh
{grover, scottm, donnlan}@ed.ac.uk

[†]Institute for Informatics and Telecommunications,
National Centre for Scientific Research “Demokritos”
{vangelis, dfarmak, samarita, petasis}@iit.demokritos.gr

[◇]D.I.S.P., Universita di Roma Tor Vergata
{pazienza, vindigni}@info.uniroma2.it

[‡]Informatique-CDC, Groupe Caisse des Depots
{frantz.vichot, francis.wolinski}@caissedesdepots.fr

Abstract

We describe the multilingual Named Entity Recognition and Classification (NERC) subpart of an e-retail product comparison system which is currently under development as part of the EU-funded project CROSSMARC. The system must be rapidly extensible, both to new languages and new domains. To achieve this aim we use XML as our common exchange format 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 preliminary evaluation results demonstrate the viability of our approach.

1. Introduction

We describe the multilingual Named Entity Recognition and Classification (NERC) component of an e-retail product comparison system which is currently under development as part of the EU-funded project, CROSSMARC. The project applies state-of-the-art language engineering tools and techniques to achieve commercial-strength information extraction from web pages. The core components of the prototype system are:

- spider agents which visit websites and return pages which are likely to contain product descriptions;
- 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.

Figure 1 shows the overall processing stages of the CROSSMARC system. We focus in this paper on the NERC subcomponent. Although NERC is a familiar task to the IE research community, there are novel aspects arising from our application area which present significant and interesting challenges both for implementation and for evaluation.

2. 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.

2.1. Multilinguality

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 multilingual 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.

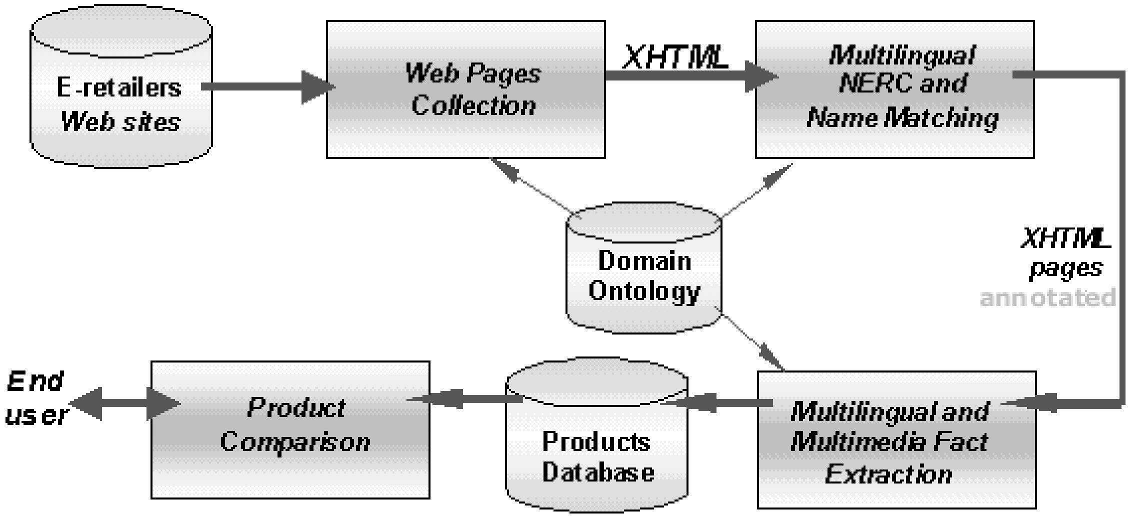


Fig. 1 CROSSMARC Processing Stages

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 (Grover et al., 2000; Thompson et al., 1997) as well as the *xmlperl* program (McKelvie, 1999). The system architecture is similar to the hybrid system which was the LTG's contribution to the MUC-7 competition (Mikheev et al., 1998): 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 specific rule sets at specific sub-parts of the XHTML tree structure.

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 (Wolinski et al., 1998; Wolinski and Vichot, 2001).

FNERC has been implemented as a standalone module in order to ease its integration into the CROSSMARC multilingual platform. It takes as input XHTML pages produced by Tidy from HTML pages. Due to the wide variety of different web sites, some functions are applied to reduce the high heterogeneity of the corpus and to normalize it for the following part of the processing. These steps are 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.

Localisation 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 (Petasis et al., 2002; Farmakiotou et al., 20002). 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 (Farmakiotou et al., 2000) 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, TIMEX. 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 (Pazienza and Vindigni, 2000) 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 utilised 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. Other components, such as an Italian part-of-speech tagger, a chunker and a shallow parser (Basili et al., 1999) 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.

2.2. 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 are shown in Table 1.

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

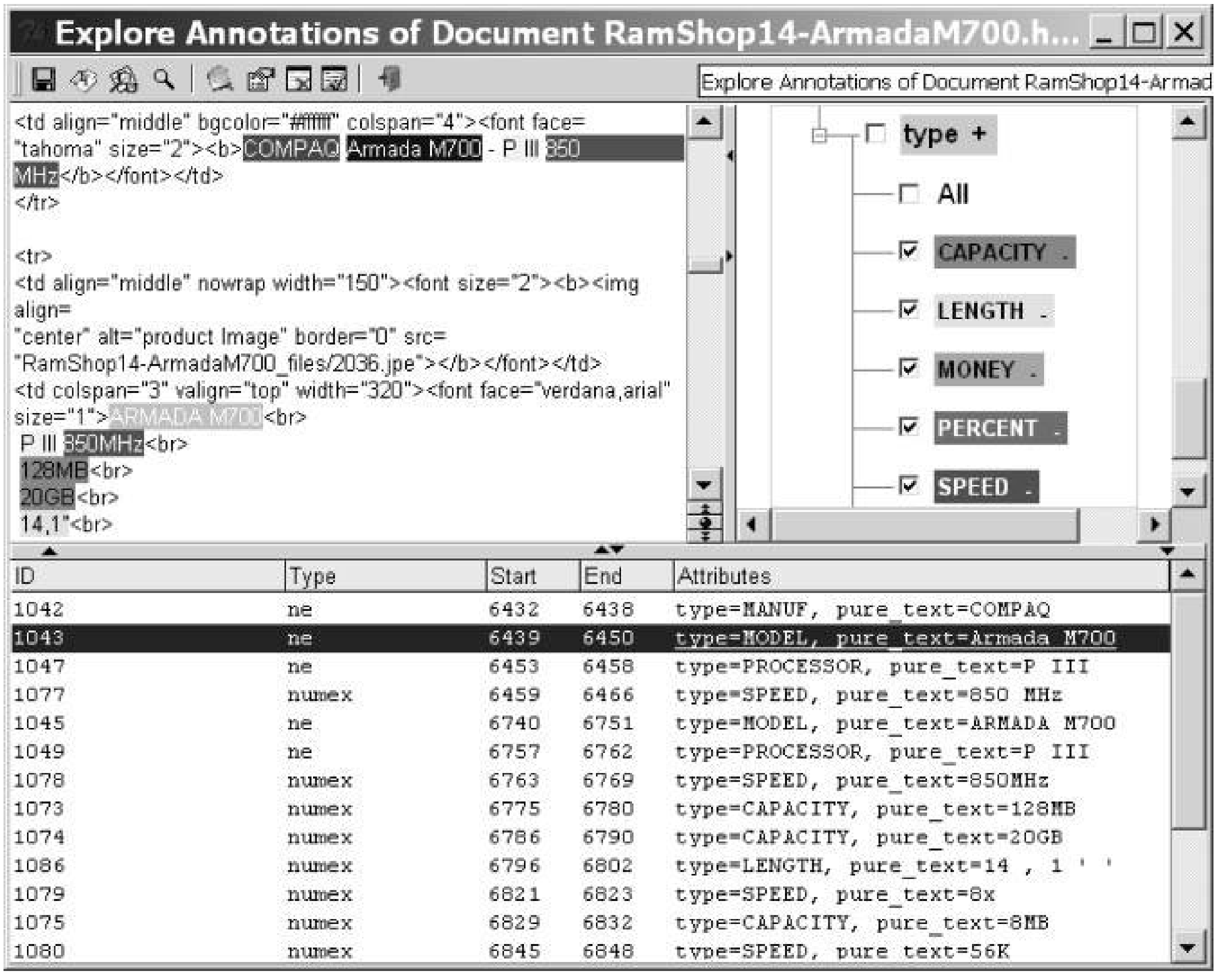
Table 1: Minimum Set of NERC Entities

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.

Figure 2 illustrates some of the entities in Table 1 by showing an example of output from the HNERC module. The HTML file is displayed in the upper left window with the entities which have been recognised highlighted in different colours. The lower window records the byte positions of the entities in the Tipster-style used by HNERC. This example show clearly how the text to be annotated occurs deeply nested inside the HTML structure.

We view the set of entities in Table 1 as a minimum set of requirements on the individual NERC components. They represent the most salient characteristics of laptops and they are the entities that occur most consistently and most frequently in product descriptions 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 Table 1 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.



Explore Annotations of Document RamShop14-ArmadaM700.h...

```
<td align="middle" bgcolor="#ffffff" colspan="4"><font face=
"tahoma" size="2"><b>COMPAQ Armada M700 - P III 850
MHz</b></font></td>
</tr>

<tr>
<td align="middle" nowrap width="150"><font size="2"><b><img
align=
"center" alt="product Image" border="0" src=
"RamShop14-ArmadaM700_files/2036.jpe"></b></font></td>
<td colspan="3" valign="top" width="320"><font face="verdana,arial"
size="1">ARMADA M700<br>
 P III 850MHz<br>
 128MB<br>
 20GB<br>
 14,1"<br>
```

ID	Type	Start	End	Attributes
1042	ne	6432	6438	type=MANUF, pure_text=COMPAQ
1043	ne	6439	6450	type=MODEL, pure_text=Armada M700
1047	ne	6453	6458	type=PROCESSOR, pure_text=P III
1077	numex	6459	6466	type=SPEED, pure_text=850 MHz
1045	ne	6740	6751	type=MODEL, pure_text=ARMADA M700
1049	ne	6757	6762	type=PROCESSOR, pure_text=P III
1078	numex	6763	6769	type=SPEED, pure_text=850MHz
1073	numex	6775	6780	type=CAPACITY, pure_text=128MB
1074	numex	6786	6790	type=CAPACITY, pure_text=20GB
1086	numex	6796	6802	type=LENGTH, pure_text=14 , 1 ' '
1079	numex	6821	6823	type=SPEED, pure_text=8x
1075	numex	6829	6832	type=CAPACITY, pure_text=8MB
1080	numex	6845	6848	type=SPEED, pure_text=56K

Figure 2. Screenshot of HNERC

3. Working with Web Pages

While the CROSSMARC partners all have a strong background in NERC in domains which use flat, discursive text (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. In addition to text, they also contain links, images and buttons. Statistical corpus analysis has shown that web page language forms a distinct genre and that it follows separate grammar, paragraph and sentence formation rules and conventions. These kinds of differences can affect the performance of standard NLP techniques when transferred to web pages (Amitay, 1997; Soderland, 1997). An informal comparison of a corpus of web pages and flat texts of the same domain (descriptions of laptops coming from computer magazines) in the context of CROSSMARC showed the following:

- Paragraphs and sentences in web pages are usually much shorter than ones frequently encountered in free text.
- Itemized lists and tabular format are used more frequently in web pages 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, e.g. *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, which is processed word by word, the processing of web pages must take into account the fact that the 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. The fact that HTML documents are frequently far from well-formed imposes greater difficulty in their processing and makes the use of programs such as Tidy[1] imperative for the production of well-formed (XHTML) pages.

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

HTML tags have been used as an exclusive means for name recognition and identification in the creation of wrappers (Kushmerick, 1997). The most common approach to extracting information from the web is the training of wrappers using wrapper induction techniques (Kushmerick et al., 1997; Muslea et al., 1998). The disadvantage of this method is that it requires large quantities of manually tagged training data and, moreover, it can only be applied successfully to pages that have a very rigid format. In CROSSMARC we attempt to balance the use of HTML layout information with the use of linguistic information in order to enable NERC and Fact Extraction 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 to a representation that enables the use of linguistic processing.

4. Localization

In a multilingual 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*, Ιόνιον Λιϑίον. 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.

A further task that must be addressed to aid comparison across products concerns the normalization of measurement expressions, so that two descriptions of similar things use a canonical unit of measurement. For example, hard disk capacity can be measured in Gigabytes or Megabytes but to facilitate comparison it is useful to encode a conversion to the canonical unit as an XML attribute value.

5. Evaluation

For the creation of the necessary training and testing corpora, the Greek partners developed a "Web Annotator" tool which is designed to assist human annotators in tagging the named entities found in the web pages. The tool takes as input the NERC DTD for the relevant domain (e.g. the laptop 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 was collected for each language and the Web Annotator was used to add annotations for the entities described in Table 1. Tags to identify the boundaries of each distinct product description in the page were also added.

The four language corpora 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 2.

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

Table 2: The Test Corpora

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 2 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 Information Extraction 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 favour 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 3 we average across all four monolingual systems to provide evaluation results for version 1 of the CROSSMARC NERC system as a whole[2].

		Prec	*Rec*	*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	0.92	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 3: Evaluation Results

From Table 3 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 favourably 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 judgement 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 upcoming 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.

6. Related Work

Within the field of Information Extraction 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) competitions have been a highly visible forum for reporting IE and NERC results, see for example, MUC-7 (Chinchor, 1998). The systems participating in MUCs are required to process newspaper 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 manually the templates, 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 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 statistical, with hybrid combined rule-based and statistical systems 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 in the competition were 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, Califf, 1998; Freitag and Mccallum, 1999; Soderland, 1999). The alternative to this is to use machine learning over unannotated data (Collins and Singer, 1999; Riloff and Jones, 1999; Yangarber and Grishman, 2000; Yangarber et al., 2000).

There has been increasing interest in the web either just as a source of data or, as in the case of CROSSMARC, as an application area for IE technology. The most common approach to extracting information from the web is wrapper induction (Kushmerick, 1997; Kushmerick et al., 1997; Muslea et al., 1998). However, this method can only be successfully applied to pages that have a standardised format and not pages that present a more irregular format

[2] We have not included figures for the PERCENT subtype of NUMEX or the TIME subtype of TIMEX because of sparse data. The figures in the ALL row are averaged from the raw data rather than from the figures in the other rows.

(for relevant experiments see Soderland, 1997). In addition, it requires a very large quantity of manually tagged training data.

In the field of multilingual Information Extraction, there are three main approaches (Azzam et al., 1999), which we itemize below in the context of a hypothetical system which works with English and Italian and which presents information to an English user:

- a full Italian-English MT system translates all the Italian texts to English. The English IE system is then used to extract the information from the translated texts and present it to the user;
- two local (English, Italian) IE systems process the texts in the two languages. A mini Italian-English MT system is then used to translate the information extracted by the Italian IE system;
- a general IE system uses two local (English, Italian) syntactic/semantic analysers and a language independent domain model to produce a language independent representation of the information extracted from the English and Italian texts. English lexical resources are then used to generate in English the extracted information from the domain model.

The IE approach presented in (Kameyama, 1997) belongs in the second category whereas the approach presented in (Azzam et al., 1999) is in the third category. The CROSSMARC approach belongs in the second category of systems, although we do not use an MT system to translate the information extracted but, instead, we exploit the ontology and the corresponding lexicons.

7. Acknowledgements

CROSSMARC is a project of the Information Society Technologies (IST) Programme of the European Union (IST 2000-25366). The CROSSMARC partners are NCSR "Demokritos" (Greece, coordinator), University of Edinburgh (UK), Universita di Roma Tor Vergata (Italy), Informatique CDC (France), and VeltiNet (Greece). Further information about CROSSMARC can be found at the project's website, `http://www.iit.demokritos.gr/skel/crossmarc/`.

8. References

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

S. Azzam, K. Humphreys, and R. Gaizauskas. 1999. Using a language independent domain model for multilingual information extraction. *Applied Artificial Intelligence (AAI)*, 13(6).

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.

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

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`.

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*.

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.

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.

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.

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`.

M. Kameyama. 1997. Information extraction across linguistic barriers. In *Proceedings of the 1997 AAAI Symposium on Cross-Language and Speech Retrieval*, Stanford, CA. Stanford University.

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.

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

D. McKelvie. 1999. XMLPERL 1.0.4. XML processing software. `http://www.cogsci.ed.ac.uk/~dmck/xmlperl`.

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`.

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

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

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.

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, FL.

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.

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

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/`.

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

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

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)*.

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