2nd Hellenic Conf. on AI, SETN-2002, 11-12 April 2002, Thessaloniki, Greece,
Proceedings, Companion Volume, pp. 91-102

Named Entity Recognition in Greek Web Pages

Dimitra Farmakiotou, Vangelis Karkaletsis, Georgios Samaritakis, Georgios Petasis and Constantine D. Spyropoulos

Software and Knowledge Engineering Laboratory
Institute of Informatics and Telecommunications, N.C.S.R. "Demokritos"
{dfarmak, vangelis, samarita, petasis, costass}@iit.demokritos.gr

Abstract. We describe the functionalities of the Hellenic Named Entity Recognition and Classification (HNERC) system developed in the context of the CROSSMARC project. CROSSMARC is developing technology for e-retail product comparison. The CROSSMARC system locates relevant retailers' web pages and processes them in order to extract information about their products (e.g. technical features, prices). CROSSMARC's technology is demonstrated and evaluated for two different product types and four languages (English, Greek, Italian, French). This paper presents the HNERC system that is responsible for the identification and classification of specific types of proper names (e.g. laptop manufacturers, models), numerical expressions (e.g. length, weight), and temporal expressions (e.g. time, date) in Hellenic vendor sites. The paper presents the HNERC processing stages using examples from the laptops domain.

1 Introduction

Named-entity recognition and classification (NERC) is the identification of proper names, numerical and temporal expressions in text and their classification as different types of named entity (NE), e.g. person and organisation names in financial news, or names of manufacturers and models in web pages that contain descriptions of computer goods. NERC is an important subtask in most language engineering applications, in particular information retrieval and information extraction.

Existing NERC systems belong to the following broad categories:

- Systems based on hand crafted grammars and gazetteers. Typical examples are LaSIE II [8] and FASTUS [2]
- Systems exploiting Machine Learning techniques for the automatic acquisition of NERC lexical resources. MENE [4] and Nymble [3] are examples of such systems.
- Systems combining the two previous approaches like the LTG system of the University of Edinburgh [13].

In this paper, we describe the functionalities of the Hellenic NERC (HNERC) system developed in the context of the CROSSMARC[1] project. CROSSMARC is developing technology for e-retail product comparison. The CROSSMARC system locates retailers' web pages and processes them in order to extract information about their products. CROSSMARC's technology is demonstrated and evaluated for two different product types and four languages (English, Greek, Italian, French). The HNERC system is responsible for the identification and classification of specific types of proper names (e.g. laptop manufacturers, models), numerical expressions (e.g. length, weight), and temporal expressions (e.g. time, date) in Hellenic vendor sites. HNERC involves first a lexical preprocessing stage and a Gazetteer lookup stage. Handcrafted pattern grammars are applied against the resulting representation of these stages for the identification and classification of named entities.

Section 2 provides information on related work. Section 3 discusses the differences between raw text and hypertext. The distinct characteristics of hypertext require the adaptation of NERC systems that operate only in raw text if they are to process web pages instead. This is the case for HNERC, which is based on a NERC system that had been developed for the processing of raw text. The HNERC system is described in Section 4. An initial evaluation of HNERC is presented and discussed in Section 5. The paper concludes with our future plans in Section 6.

2 Related Work

Recent progress in information extraction (IE) technology is due to the increase in available resources such as machine-readable dictionaries and text corpora, in computational power and processing volume as well as the development of Language Technology techniques that can be applied in practice. This progress is evident in the results of the Message Understanding Conferences[2] (MUCs) in which several IE systems have been evaluated. NERC is the IE evaluation task for which the best results have been achieved, proving that this technology is mature. The systems participating in MUCs are required to process 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. At recent MUCs, English named-entity recognizers reached performance comparable to that of humans ([13], [11]). At the same time, researchers across Europe and elsewhere have developed named-entity recognizers for several languages other than English (e.g. [10], [15], [20], [22]).

---

[1] CROSSMARC (IST 2000 – 25366) is a 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), ICN (France). http://www.iit.demokritos.gr/skel/crossmarc/

[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

NERC systems typically exploit lexicons and grammars, which need adaptation whenever a system is customized to a new domain. Manual construction and adaptation of these resources is a time-consuming process and it is therefore worth examining methods that could automate their construction. The exploitation of learning techniques to support the customization of NERC systems has recently attracted a lot of attention. Nymble [3] and Alembic [6] are examples of systems exploiting learning techniques for NERC systems.

In the Software & Knowledge Engineering Laboratory (SKEL) a number of different techniques have been examined for the development and evaluation of Greek NERC systems. Systems exploiting handcrafted grammars have been developed for the domain of management succession news [10] and Stock Market news [7]. Machine learning techniques have also been used in these two domains as well as other domains ([17], [18], [19]). All these systems process free text. CROSSMARC is the first SKEL project that concerns the processing of Web pages, a document type with different requirements that led us to adapt our existing NERC technology.

Existing systems that extract data from Web pages are called wrappers and in most cases they are not based on linguistic knowledge, but they mainly use delimiter-based extraction patterns. The development of wrappers is a time consuming process, since web pages are constantly changing. For this purpose, the development of techniques for automatically generating wrappers, i.e. wrapper induction using *inductive learning*, is necessary. WIEN [12] and STALKER [14] are examples of wrapper induction systems. Their drawback is that they can be successfully applied to pages that have a standardized format rather than pages that present a more "irregular" format and require vast numbers of manually tagged training data.

In the CROSSMARC project, we aim at the exploitation of linguistic knowledge for the extraction of information from web pages, adapting our existing NERC systems that operate only in raw texts.

3. From raw text to Web pages

The Hellenic NERC system (HNERC) that is being developed in the context of CROSSMARC is based on a previous version of the NCSR NERC system (MITOS NERC) that operates only in raw text and not in web pages. MITOS NERC needed adaptation in order to take into account not only the characteristics of the CROSSMARC domains but also the genre of hypertext it processes. In this section, we outline the processing stages of MITOS NERC and describe the requirements imposed for the processing of web pages.

3.1 Named Entity Recognition in Raw Text: the MITOS NERC

The MITOS NERC system [7] forms part of a larger Greek information extraction system, that was developed in the context of the R&D project MITOS[3] [9]. MITOS NERC recognizes names of Organizations, Persons and Locations from free text.

MITOS NERC consists of three processing stages: linguistic pre-processing, NE identification and NE classification. The linguistic pre-processing stage involves the following tasks: tokenization, sentence splitting, part-of-speech tagging and stemming.

The NE identification stage involves the detection of the start and the end of all possible spans of tokens that are likely to belong to a NE. Identification consists of three sub-stages: initial delimitation, separation and exclusion. Initial delimitation involves the application of general patterns that are combinations of a limited number of words, selected types of tokens (e.g. capitalized words), symbols and punctuation marks. At the separation sub-stage, possible NEs that are likely to contain more than one NE or a NE attached to a non-NE are detected and attachment problems are resolved. Finally, at the exclusion sub-stage the context of a NE and membership in exclusion lists are the criteria used for exclusion from the possible NE list. Suggestive context for exclusion consists of common names that refer to products, services or artifacts.

The classification of the identified NEs involves three sub-stages: application of classification rules, gazetteer-based classification, and partial matching of classified named-entities with unclassified ones. Classification rules take into account both internal and external evidence, i.e., the words and symbols that comprise a possible name and the context in which it occurs. Gazetteer-based classification involves the look up of pre-stored lists of known proper names (gazetteers). At the partial matching sub-stage, classified names are matched against unclassified ones aiming at the recognition of the truncated or variable forms of names.

3.2 From Raw Text to Web Pages: The new text genre and domain

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], [21].

An informal comparison of a corpus of Web pages to flat texts of the same domain (descriptions of laptops coming 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

[3] http://www.iit.demokritos.gr/skel/mitos

- 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 of other possible names or expressions of interest.
- Unlike flat text that is processed word after 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. 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. Subsequently, 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 many times far from well formed imposes greater difficulty in their processing and makes the use of programs like Tidy[4] imperative for the production of 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 [12]). Their drawbacks are that they require vast numbers of manually tagged training data and that they can be successfully applied to pages that have a rigid format rather than pages that present a more "irregular" format (for relevant experiments see [21]). Our approach, which is described in the next section, 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.

4. Hellenic NERC in CROSSMARC

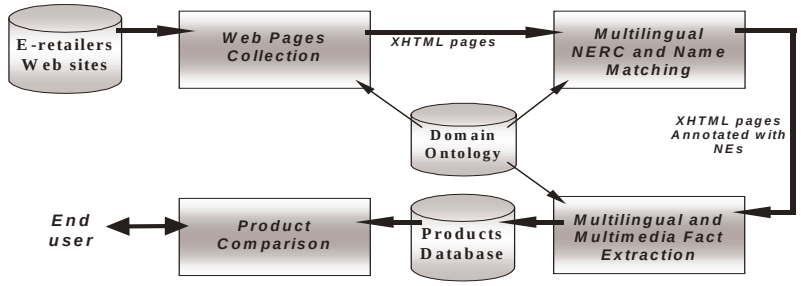
HNERC aims at the identification and classification of specific types of proper names, numerical expressions and temporal expressions in Hellenic vendor sites. HNERC

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

exploits the facilities provided by the Ellogon text engineering platform [16] and is based on the MITOS NERC briefly described in 3.1 [7].

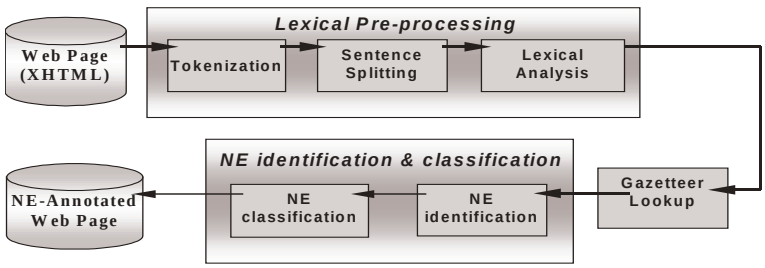
HNERC, like every language-specific NERC system in CROSSMARC, processes the output of the module responsible for the collection of web pages in the product domain (laptops is the first product domain) and sends its output to the fact extraction module that is responsible for extracting product information which then fill in a database of product descriptions (Figure 1). The output of HNERC is the page that contains laptop descriptions in which the recognized and classified named entities are annotated with the XML tags specified in the CROSSMARC NERC Document Type Definition (DTD), which is shared by all language specific modules in CROSSMARC.

Fig. 1. CROSSMARC Architecture



The overall structure of HNERC is depicted in Figure 2. The main processing stages of HNERC are described in the following subsections.

Fig. 2. The Hellenic NERC Architecture



4.1 Lexical Pre-processing

The tokenizer reads the XHTML document maintaining layout information in order to split the document into zones of interest (titles, paragraphs, tables, lists, images). This is not necessary for raw text. However, in the case of web pages, the exploitation of

layout information for the identification of zones of interest is necessary since different zones require different processing. A paragraph usually consists of periods separated by punctuation marks, whereas text contained in a table cell or a list element, usually consists of only a few words and it may not make sense without text contained in other cells of the same table. The tokenizer separates the text into tokens, annotates them with a predefined set of tags, and separates the text into zones depending on the types of tokens recognized. In the following example

```
<table> <tbody>
<tr> <td> <img src=""> <br>  <b> TOSHIBA 2800-600 </b>
</td> </tr>
<tr> </td> <td> <b> Επεξεργαστής: </b></td> <td>INTEL
PIII 1000</td> </tr>
<tr> <td><b>Οθόνη:</b></td> <td>14.1 inches TFT</td>
</tr> </tbody> </table>
```

the tokenizer outputs the page source as HTML tokens (*<table>*, *<tbody>*, *<tr>*, *<td>*, *<img src="">*, *
*, **, **, *</td>*, *</tr>*, *</tbody>*, *</table>*), Latin Word tokens (*TOSHIBA*, *INTEL*, *PIII*, *TFT*, *inches*), Greek word tokens starting with an uppercase character (*Επεξεργαστής* (Processor), *Οθόνη* (Screen)), Numbers (*2800*, *600*, *14*, *1*), symbols (–), punctuation marks (*:*, .), images (*<img src="">*). The whole example is tagged as a table zone and the textual (non HTML) content of each cell is marked. "*TOSHIBA 2800-600*" is the content of cell 1.1, cell 2.1 is comprised by "*Επεξεργαστής:*", the text content of cell 2.2 is "*INTEL PIII 1000*" etc. These types of information are useful to the tools that follow.

The sentence splitter performs some kind of "sentence" splitting depending on the type of zone. For example, a paragraph may be split into sentences with each sentence starting after a full stop and a table may be split into "sentences" if we treat each cell or each row as a special type of sentence. Sentence annotations are added as well as attributes, such as the constituents of a sentence (i.e., IDs of the tokens that comprise a sentence) or the type of the sentence (e.g. paragraph sentence, cell sentence, row sentence etc.). Thus, in the page source example "*TOSHIBA 2800-600*" is tagged as a cell and as a row sentence, "*Επεξεργαστής*:" and "*INTEL PIII 1000*" comprise two distinct cell sentences, and "*Επεξεργαστής*: *INTEL PIII 1000*" comprises a single row sentence. In this manner text layout information is taken into consideration and can be exploited at a later stage for the classification of named entities.

The lexical analyzer has been the only module that did not require any serious form of adaptation for processing hypertext apart from taking as input the output of the sentence splitter. It exploits the results of a machine learning based part of speech tagger and a morphological analyzer.

4.2 Gazetteer Lookup

The Gazetteer Lookup tool comprises of gazetteers for terms and names. The gazetteer lookup adds annotations to those words/phrases that belong to its gazetteers. At present the size of the gazetteers is rather small, there are 706 entries for names and

terms in total. The tool has been adapted for the new genre so as to operate only in those parts of the text that comprise sentences (i.e. so as not to annotate names that are likely to exist in names of images or JavaScript code). Following the page source example in section 4.1 the tool tags "*TOSHIBA*" as a laptop manufacturer's name, "*INTEL PIII*" as a processor name and "*TFT*" as a term referring to a type of screen.

4.3 Identification and Classification of Named Entities

Pure text from a web page source, i.e. text contained in sentences without HTML tags, and the relevant pieces of information provided by the lexical preprocessing and gazetteer lookup stages are transformed into an internal representation. This is one more adaptation since raw text did not present such problems. Pattern grammars are applied against the representation for the identification and classification of named entities.

There are two types of identification patterns, patterns that identify and classify names or expressions at once and patterns that only identify possible names and expressions of interest leaving the classification problem to classification patterns. Following the example in Section 4.1, numeric expressions with unambiguous measurement units e.g. "*14.1 inches*" can be identified and classified by an appropriate pattern at once. On the contrary names like "*2800-600*" and numeric expressions that appear without measurement units, e.g. "*1000*" are first identified by identification patterns that detect the start and end of all token sequences that are likely to constitute named entities. Token sequences that have been tagged by the gazetteer as names are also considered possible entity names (PNEs), since more criteria are required for their classification. The identified PNEs are further processed by stricter identification patterns that aim at the "correction" of the initial delimitation output. These patterns are used for dropping out whole PNEs or parts of PNEs (e.g. names that do not belong to the classes of interest, words/phrases mistakenly identified as part of a PNE), as well as for separating two distinct NEs inside a PNE. These stricter identification patterns may also classify certain types of NEs.

Classification of NEs is performed in two different sub-stages. At the first sub-stage pattern grammars that combine information about internal and external evidence are used along with information from the gazetteer. "*TOSHIBA*" is classified as manufacturer by a pattern specifying that a manufacturer name must be a capitalized word, annotated by the gazetteer tool as such. "*INTEL PIII*" is classified as processor by a pattern using membership in the processor gazetteer and capitalization criteria. The second classification sub-stage uses the already classified entities as information from context and takes into account the proximity of certain types of classified NEs to unclassified ones for the classification of the latter. "*2800-600*" is recognized as a model name since it follows "*TOSHIBA*" that has been classified as a manufacturer's name. In the same manner, "*1000*" is classified as processor speed since "*INTEL PIII*" that precedes it has been classified as processor.

5. An Initial Experiment

This section describes the process followed for the collection of training and testing corpora, their annotation and discusses the results of an initial experiment.

5.1 Training and Testing Corpora

In the context of CROSSMARC, four sets of corpora (one for each language) have been compiled for system development and evaluation purposes. The Hellenic corpora have been separated manually into training and testing corpora with care that the testing corpus contains a considerable number of pages from sites that are not present in the training corpus.

For the annotation of the corpora for all four languages the human annotators followed Guidelines that had been specifically issued for the domain of laptop products following the example set by the MUCs [5]. Annotation guidelines are useful to human annotators and system developers alike, since they specify the outcome of the manual annotation and the desired NERC system output with directives and appropriate examples. The CROSSMARC Annotation Guidelines define the types of names and expressions to be annotated and attempt to determine the boundaries of names and expressions within a text. These boundaries are not always straightforward, a processor's name, for instance, within the text string "*Intel Mobile Pentium III Processor*" can be any of the following strings if relevant annotation directions are not available: "*Intel Mobile Pentium III Processor*", "*Intel Mobile Pentium III*", "*Mobile Pentium III*", "*Pentium III*". The directions aim at a uniform annotation of the corpora not only in the case of various tagging possibilities of full name forms but also in the cases of names or expressions occurring within elliptical constructions (e.g. "*Windows 98 and 2000*"), numeric ranges (e.g. "*1-3 days*"), etc.

The human annotators used a Web Page Annotation Tool that allows users to tag web pages by selecting text strings and clicking on relative tags. The same annotation process for the creation of the training and testing corpus is followed for all four languages in CROSSMARC.

5.2 Evaluation Results

An initial system evaluation has been performed on a subset of the testing corpus that comprises of 31 pages with 32 laptop descriptions. The experiment was conducted twice, the first time for four types of named entities Manufacturer (Manuf), Operating System (OS) / Software, Laptop Model and Processor and the second time for seven out of the fifteen types of Named Entities (Manuf, Laptop Model, Processor, Operating System (OS) / Software, Money, Speed and Capacity). For the measurement of system performance the metrics of Recall, Precision and F-measure have been used. Recall measures the number of items of a certain named entity type correctly identified, divided by the total number of items of this type. Precision is the ratio of the number of items of a certain named entity type correctly identified to all

items that were assigned that particular type by the system. F-measure combines Recall (R) and Precision (P) using the formula ((2xPxR)/(P+R)).

The first experiment pointed to problems in the preprocessing stages, (tokenization and sentence-splitting). Text zones had been erroneously detected within HTML meta-content, or JavaScript code. Thus a considerable number of Manufacturer, Model and Processor names had been recognized in parts of the page source that are not visible in a browser. Consequently, Precision was rather low even for categories like Manuf (0.538) and Processor (0.540), which comprise small sets of names.

The second experiment was conducted when tokenization and sentence splitting problems had been solved and the pattern grammars had been extended to more types of NEs. The results of the second experiment (Table 1) are encouraging for most NE types. Low Recall in the OS/Software category is not much of a surprise since it is an open class of names and many laptop manufacturers present software specially designed for their own products. This also explains a precision of 0.781 in the Manuf category as software produced by laptop manufacturers is named after them, e.g. "Sony Notebook Setup", presenting us with a disambiguation problem that we will attempt to solve with appropriate patterns.

Table 1. Evaluation Results from the Second Experiment

	Manuf	*Model*	*Processor*	*OS / Software*	*Money*	*Speed*	*Capacity*
Precision	0.781	0.871	0.933	0.840	0.887	0.983	0.848
Recall	0.961	0.850	0.875	0.567	0.916	0.875	0.884
F-measure	0.862	0.860	0.903	0.677	0.901	0.926	0.865

6. Concluding Remarks

The HNERC system presented in this paper employs linguistic and layout information for the recognition of Named Entities in web pages. It has been based on a system that operates in raw text and therefore needed adaptation. We studied the differences between web pages and raw text in order to decide on the adaptations that should be performed. Apart from the domain-specific parts (gazetteers and grammar), that are adapted whenever a new domain is faced, we had also to adapt some of the domain independent modules in order to take into account the characteristics of the web pages. The modular architecture of the MITOS NERC system facilitated the adaptation task allowing the development of the first versions of HNERC in a rather short period.

System evaluation presented encouraging results that can be improved further so that the resulting technology can be commercially exploitable. Our next step is the combination of the hand crafted NERC system with one created using machine learning techniques. We have already started experimenting with the development of a NERC system that exploits supervised learning techniques.

The final NERC system will be integrated in the complete IE system, which is currently being developed in the context of the CROSSMARC project. Our aim is to develop a general purpose NERC system that can be easily customized to a new domain taking into account not only the linguistic characteristics of the domain but also the stylistic ones as it is the case with the web pages in CROSSMARC.

References

1. Amitay B.: Hypertext: The Importance of Being Different. MSc Dissertation, Univ. of Edinburgh (1997)
2. Appelt D., Hobbs J. R., Bear J., Israel D., Kameyama M., Kehler A., Martin D., Myers K., Tyson M.: SRI International FASTUS system MUC-6 Test Results and Analysis. Proc. of the 6th Message Understanding Conference (1995)
3. Bikel D., Miller S., Schwartz R., Weischedel R.: Nymble: A High-Performance Learning Name-finder. Proc. of the 5th Conference on Applied Natural Language Processing (1997)
4. Borthwick J., Sterling E., Agichtein A., Grishman R.: "NYU: Description of the MENE Named Entity System as Used in MUC-7" Proc. of the 6th Message Understanding Conference (1998)
5. Chinchor N.: MUC-7 Named-Entity Task Definition Version 3.5, (1997) http://www.itl.nist.gov/iad/894.02/related_projects/muc/proceedings/muc_7_proceedings/overview.html
6. Day D., Robinson P., Vilain M., Yeh A.: Description of the ALEMBIC System as Used for MUC-7. Proceedings of the 7th Message Understanding Conference, Fairfax, Virginia http://www.muc.saic.com (1998)
7. Farmakiotou D., Karkaletsis V., Koutsias J., Sigletos G., Spyropoulos C.D., Stamatopoulos P.: "Rule-based Named Entity Recognition for Greek Financial Texts", Proceedings of the Workshop on Computational lexicography and Multimedia Dictionaries (COMLEX 2000), , Patras, Greece (2000) 75-78
8. Humphreys K., Gaizauskas R., Azzam S., Huyck C., Mitchell B., Cunningham H., Wilks Y.: University of Sheffield: Description of the LaSIE-II System as Used for MUC-7. Proc. of the 7th Message Understanding Conference (1998)
9. Karkaletsis V., Farmakiotou D., Koutsias J., Paliouras G., Androutsopoulos I. Petasis G., Spyropoulos C.: Information Extraction from Greek Texts in the MITOS Text Management System, NCSR "Demokritos", Internal Report (2001)
10. Karkaletsis V., Paliouras G., Petasis G., Manousopoulou N., Spyropoulos C.: Named-Entity Recognition from Greek and English Texts. Journal of Intelligent and Robotic Systems 26 (1999) 123-135
11. Krupka G. R. and Hausman K.: `IsoQuest Inc.: Description of the NetOwl extractor system as used for MUC-7', in Proceedings of the Seventh Message Understanding Conference (1998)
12. Kushmerick N.: Wrapper Induction for Information Extraction, PhD Thesis, U. of Washington (1997)

13. Mikheev A., Grover C., Moens M.: Description of the LTG System Used for MUC-7" http://muc.saic.com/proceedings/muc_7_toc.html (1998)
14. Muslea I., Minton S., Knoblock C.: STALKER: Learning extraction rules for semistructured Web-based information sources. AAAI-98, Madison, Wisconsin (1998)
15. Pazienza M.T., Vindigni M.: Identification and classification of Italian Complex Proper Names. In Proceedings of the ACIDCA2000 International Conference, Tunisia (2000)
16. Petasis G., Karkaletsis V., Paliouras G., Spyropoulos, C.D.: "Ellogon: A Text Engineering Platform", NCSR "Demokritos", Internal Report (2001)
17. Petasis G., Petridis S., Paliouras G., Karkaletsis V., Perantonis S., Spyropoulos C.D.: "Symbolic and Neural Learning of Named-Entity Recognition and Classification Systems in Two Languages," In Advances in Computational Intelligence and Learning: Methods and Applications, H-J. Zimmermann, G. Tselentis, M. van Someren and G. Dounias (eds), Kluwer Academic Publishers (2001)
18. Petasis G., Vichot F., Wolinski F., G. Paliouras, V. Karkaletsis, Spyropoulos C.D.: " Using Machine Learning to Maintain Rule-based Named-Entity Recognition and Classification Systems ". Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Toulouse (2001) 426-433
19. Petasis G., Cucchiarelli A., Velardi P., Paliouras G., Karkaletsis V., Spyropoulos C.D.: "Automatic adaptation of proper noun dictionaries through co-operation of machine learning and probabilistic methods". Proc. of the 23rd ACM SIGIR Conference on R&D in IR (SIGIR), Athens, Greece (2000)
20. Piskorski J., Neumann G.: An Intelligent Text Extraction and Navigation System. Proceedings of the 6th International Conference on Computer-Assisted Information Retrieval (RIAO-2000) Paris (2000)
21. Soderland S.: Learning to extract text-based information from the World Wide Web, in Proc. of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD-97) (1997) 251-254
22. Vichot F., Wolinski F., Ferri H.C., Urbani D.: Using Information Extraction for Knowledge Entering. In S. Tzafestas (Ed.) Advances in Intelligent Systems: Concepts, Tools and Applications, Kluwer Academic Publishers (1999)