# Adaptive, Multilingual Named Entity Recognition in Web Pages

## C0702

**Abstract.** The identification of interesting web sites and web pages together with the extraction of information from them is a very interesting but complex task. Several commercial systems are trying to implement robust methodologies for extracting information of interest for the final user. Most of the information on the Web today is in the form of HTML documents, which are designed for presentation purposes and not for machine understanding and reasoning. The extraction task becomes even harder in a multilingual context, where web pages in different languages need to be analysed. Existing systems require a lot of human involvement for maintenance due to changes to targeted web sites and for adaptation to new web sites or even to new domains. This paper presents the adaptive, multilingual named entity recognition and classification (NERC) technologies developed for processing web pages in the context of an R&D project. The evaluation results demonstrate the viability of our approach.

## 1 INTRODUCTION

A number of systems have been developed to extract structured data from web pages. Such systems commonly include a set of wrappers that extract the relevant information from multiple web sources and a mediator that presents the extracted information in response to the users' requests (for a survey of existing web extraction tools, see [8]). Most of the existing systems use delimiter-based approaches. Documents 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 HTML tags 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 descriptions written in free text. In addition, these approaches also suffer from maintainability problems due to changes in the web sites, as well as from adaptability problems when new sites in the same domain must be analysed or when a new domain is to be processed.

These problems were the motivation for a European funded R&D project. Addressing these problems requires the combined use of techniques from the area of natural language processing (NLP) for analysing unstructured or semi-structured content in different languages, from the area of machine learning for facilitating adaptation, as well as from the area of ontology engineering for facilitating the identification, classification and normalisation of named entities (exploiting existing ontology instances and taxonomic relationships in the ontology). This project applies state-of-the-art language engineering, machine learning and ontology-based tools and techniques to develop technology for Web information retrieval and extraction. The project implemented an open, multi-lingual and multi-agent architecture integrating its components into a web-based prototype system; an infrastructure was also set up for configuring to new domains and languages [self-reference, omitted]. The prototype involves components for:

- the collection of interesting and domain-specific web pages,
- the extraction of information about product/offer descriptions from the collected web pages, and
- the storage and presentation of the extracted information to the end-user according to his/her preferences.

The customization infrastructure provides methodologies and tools for:

- the creation and maintenance of domain specific resources (ontologies, lexica),
- the collection and annotation of corpora necessary for the training and evaluation of the various components, and
- the customization of each component exploiting the domain specific resources and the collected corpora.

The system's approach to information extraction (IE) relies on a pipeline of three components: a *named entity recognition and classification (NERC) component*, a *demarcator* and a *fact extraction (FE)* component. NERC identifies domain-specific named entities in pages from different sites; the demarcator groups the identified entities into products/offers inside the page. Then, FE identifies domain-specific facts, i.e. assigns domain-specific roles to some of the entities identified by the NERC component.

**Although NERC is a familiar task within the IE research community, our work advances the state of the art as it presents a thorough evaluation of three different NERC technologies (from rule-based to hybrid to machine learning) with different adaptation strategies on two thematic domains, across four languages**. Additionally, the selection of thematic domains (laptop offers, job offers) which involve a great variety of entity types (compared for instance to news articles used in several NERC applications), along with the fact that web documents are processed instead of raw text, raised several significant and interesting challenges both for implementation and for evaluation.

Section 2 presents recent work and trends in the area of NERC, positioning our work. Section 3 describes the architecture of the multi-lingual NERC system. Section 4 discusses the customization strategies followed by the four language-specific NERC components which make up the multi-lingual NERC system. Section 5 presents the evaluation methodology and discusses the evaluation results.

## 2 RELATED WORK

NERC is considered a well-established task, as reflected in the results reported in the last Message Understanding Conferences (MUCs) [3], where some of the systems achieved performance comparable to humans. This however does not necessarily mean that NERC is an easy-to-use technology, adaptable to a wide range of thematic domains and languages. Adaptability has always been a major shortcoming for the vast majority of the NERC systems participating in MUCs, as they were either manually constructed rule-based systems or hybrid approaches, combining machine learning components with rule-based ones. As adaptability to new domains/languages has been a significant requirement for most practical NERC applications, rule-based systems lost their position as the predominant approach in favour of hybrid approaches, and, more recently, approaches based completely on machine learning. This recent interest in NERC as a machine learning task is reflected in the past two CoNLL conferences [5, 6], both of which have had language-independent NERC as their shared task. While machine learning is taking over as the predominant methodology, it is nevertheless true to say that well-crafted rule-based NERC systems are still likely to achieve better performance than the machine learning ones.

All the systems which participated in the CoNLL conferences achieved lower performance than the majority of systems in the last MUCs, despite the fact that the NERC task was easier due to the reduction in entity types and its formulation as a word tagging problem. While the CoNLL evaluations targeted multilingual NERC, no evaluation results exist regarding portability to new thematic domains. Our work presents a more thorough evaluation, as we have evaluated three different NERC technologies (rule-based, hybrid and machine learning) with different adaptation strategies (semi-automatic rule learning, resource learning and re-trainable machine learning components) on two thematic domains and across four languages. Additionally, the chosen thematic domains were not related to news articles and involved a greater variety of entity types, some of which present increased difficulty in their recognition (such as job titles in job offers).

The fact that our corpora originate from the Web makes our task even more challenging. Web pages differ from raw text in terms of content and presentation style. Apart from raw text, they also contain tables, links, images or 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].

The processing of HTML documents poses additional difficulties in comparison to plain textual documents, as new problems arise even in “trivial” pre-processing tasks. For instance, sentence boundary identification, a task which is very useful in NERC because named entities are assumed not to cross these boundaries, is far more complex on HTML documents which frequently contain either extremely short, elliptic or even ungrammatical sentences, especially since itemised lists and tabular format are used more frequently in hypertext than free text. But clearly the most important problem associated with HTML processing lies in the nature of HTML, which was designed to describe a document *visually*: elements that are visually adjacent can be very far from each other in the HTML source (for example belonging to distinct table cells), while they still belong to the same linguistic sentence. A NERC system intending to process such documents must resolve these issues in order to achieve an acceptable performance.

## 3 THE MULTILINGUAL NERC

Our multi-lingual IE integrates four language-specific sub-systems which operate as autonomous processors. A proxy mechanism, the Information Extraction Remote Invocation (IERI), was developed for interfacing with the IE sub-systems. This module takes the XHTML pages produced by the web pages collection system and routes them to the corresponding language-specific IE sub-system according to their language (see Fig. 1). The NERC components for English (ENERC), French (FNERC), Hellenic (HNERC), and Italian (INERC) are described in Section 4.

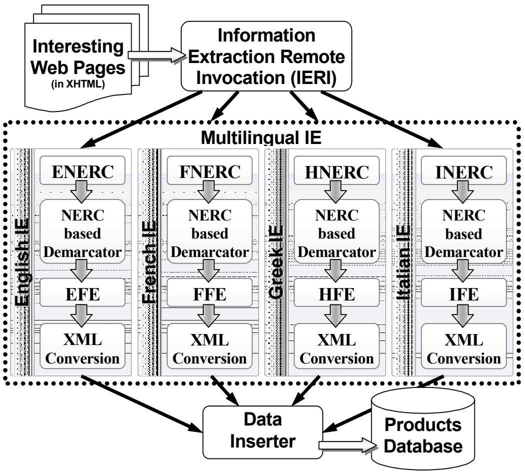

**Fig.1** The multi-lingual IE system

The architectures of these language-specific NERC components are 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 sentences, exploiting HTML tags, and some kind of word or phrase level analysis is performed. Building on this analysis results, a NERC method is applied. For each language there is a lexicon derived from the domain ontology. Each NERC component uses the appropriate lexicon in the entities identification and classification.

Although the language-specific NERC components share a similar architecture and I/O specifications, they differ in the techniques they use. All the NERC components for the 1st domain (laptops offers) were rule-based. For the 2nd domain (job offers), the researchers developing English and Hellenic systems switched to machine learning based approaches, while the Italian system remained rule-based (requiring no changes in the rule set). The French system followed a hybrid approach using machine learning techniques to induce NERC rules in a representation format that can be edited by knowledge engineers. Issues concerning adaptation to new domains have received great attention as they can speed-up the production of commercial systems. For ENERC and HNERC, strategies have been developed that center around retraining generic components on new datasets and the selection of new features. For FNERC, customization concerns the development of techniques to aid swift authoring of new rule sets. And for INERC, the focus is on customizing the lexical resources which play a key part in that system. These issues will be detailed in the next section.

# 4 EXTENSIBILITY

To support easy customization both to new domains and new languages, an ontology management system has been developed. This is based on the Protégé knowledge editor [9] and provides a set of editors and functionalities [self-reference, omitted]:

- ontology editor for the creation and maintenance of domain ontologies;
- lexicon editor for the creation and maintenance of language-specific lexicons under domain ontologies;
- NERC editor for the specification of the important entities for the domain;
- template editor for the specification of the important fact types for the domain, their relations to the NERC entities and their possible values according to the ontology;
- stereotypes editor for the creation and maintenance of the user stereotypes' definitions according to the ontology;
- functionalities for exporting the ontology and the lexicons in XML, the entities' specification as a NERC DTD, the template as an XML schema, and the stereotypes' definitions in XML.

A new domain can be easily set up with this tool, by creating the domain's ontology, the lexicons for the languages supported, the NERC DTD, the IE template, as well as the user stereotypes for providing personalized access to the extracted information.

The prototype currently includes English, French, Greek and Italian IE components, but other languages can be easily integrated. For this reason, each language-specific NERC component is only loosely coupled with the only constraints concerning input and output formats. Each language-specific NERC component takes an XHTML page as input and returns the same page augmented with XML annotations marking the named entities found in the page, according to the NERC DTD.

In a multi-lingual system questions of localization arise. 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 our system we need to ensure that we can match names that refer to the same entities across either different surface realisations or in different languages. For example, all the following names describe the same battery type: *Lithium Ion, Ions Lithium, Ioni di litio, Ιόντων Λιθίου*. Since the ontology represents the common set of concepts across all languages which play a role in the facts we aim to extract, each IE sub-system provides a pointer from extracted entities to ontology concepts. This serves not only to match different surface realisations of the same concept across languages, but also to match them within the same language.

## 4.1 English NERC Customization

The ENERC module takes a machine learning approach to named entity recognition formulating it as a word tagging problem. Two named entity taggers have been developed using the C&C tagger [4] and the openNLP maximum entropy software.[1] The customization strategy is a matter of annotating new training material and then training the classifier.

Maximum entropy, besides being a method allowing diverse pieces of contextual evidence to be incorporated, it is also well suited to language processing because it is highly effective at estimating a probability model using sparse evidence. In practice, this means that we can take a very generous approach to feature engineering where all reasonable features (that is, features that are not likely to introduce noise) are used. This makes it particularly appropriate for adaptive NERC, as even feature engineering efforts can be kept to a minimum when preparing the system for a new domain. One weakness of this approach lies in the amount of training data necessary. Maximum entropy systems for NERC tend to require large amounts of data with increased data for increased numbers of distinctions, meaning a fairly large amount of annotation needs to be done to obtain top performance. To illustrate, we performed fifty-fold cross validation on the test data. This improved significantly our evaluation results (see section 5.2).

Customizing the system to a new domain is a matter of collecting annotated training corpora and training one of the classifiers with the large feature sets provided. Of course, it may be desirable to add domain-specific features, especially where it is expensive to obtain large amounts of annotated training material. To this end, the ENERC system provides a feature engineering methodology using the openNLP software.

The ENERC approach to tokenization and part-of-speech tagging uses generic resources from the TTT XML processing tool set.[2] We have also developed a generic "sentence" identification module for web pages. As web pages do not typically contain sentences in the linguistic sense, this module approximates HTML "sentences" as sequences of words within the same HTML tag (excluding tags such as bold which occasionally are used to highlight individual words within a single logical sequence). Both of these processing resources are applicable in new domains without customization.

## 4.2 Hellenic NERC Customization

The HNERC module is built entirely on machine learning, as it uses a combination of several machine learning techniques. Its architecture can be decomposed into four subsystems. The first subsystem is responsible for performing lexical pre-processing, such as tokenisation, sentence boundary identification, part-of-speech tagging and gazetteer lookup. The customisation requirements of this subsystem are kept to a minimum: part-of-speech tagging is performed by a trainable machine learning component, which must be updated only when porting to a new language while the gazetteer lookup component is updated automatically from the training corpus and ontology lexicons. All other components are domain and language neutral, at least for the examined languages (this method was applied in all four languages of the project).

The second subsystem is the "token-based NERC": viewing NERC as a word tagging problem it operates over word tokens and applies five independent taggers, while the final tag for each word token is chosen through a simple majority voter. This subsystem is language and domain neutral. Porting into a new domain/language only requires a training corpus annotated with named entities. Once the training corpus is available, the whole subsystem can be trained through an automated process.

The third subsystem, the "phrase-based NERC", views NERC as a classification problem of phrases that possibly are names of entities. It operates over phrases which have been identified using a grammar automatically induced from the training corpus and uses a C4.5 decision tree classifier [10] to recognise which phrases

[1] http://maxent.sourceforge.net/index.html.

[2] http://www.ltg.ed.ac.uk/software/ttt/.

describe entities. Again this subsystem requires only an annotated corpus in order to be ported into new domains/languages, with its training process being fully automated.

The fourth and final subsystem combines the results of the 2nd and 3rd subsystems and performs some basic filtering over their results. Being a domain and language neutral subsystem, it requires no adaptation while porting to a new domain/language. Optionally, this subsystem can apply some additional manually developed filtering patterns. However, these patterns must be manually updated for new domains/languages, if their use is desirable.

### 4.3 French NERC Customization

The FNERC team has developed a customization methodology, which uses a tool based on machine learning techniques to help the human expert adapt the existing FNERC to a new domain. The idea here is that it is very tedious and time-consuming for a human to write easy rules, but it is very hard for a totally automatic system to write difficult rules: the solution is thus a semi-automatic customization tool. Based on the human-annotated corpus, the machine-learning module produces a first version of human-readable rules plus several useful lists of examples and counter-examples to possible rules and relevant contexts. The human expert then modifies the rule set appropriately.

The learning mechanism is applied in the same way for the left and right contexts of the entity to be recognized as well as for its beginning and end. The algorithm used is a naïve probabilistic one with rule induction using deduction of disjunction applied separately for each type of entity. As a result, the human expert receives, for each type of entity, a set of pre-evaluated rules and a set of examples and counter-examples of possible rules. The machine-learning module is based on a Logic programming approach, inspired by [7]. The results are presented to the human expert as a set of HTML files which can be easily browsed. Clicking on a word participating in a rule opens a new window with the contexts (positive and negative) containing the word. The evaluation results have shown that an equivalent level of quality can be achieved in only one third of the time needed for the development of a rule-based system.

### 4.4 Italian NERC Customization

The INERC component has a modular architecture with processing elements being general and reusable in new domains. Customization can be restricted to the knowledge bases (i.e. domain ontology, the Italian lexicon and terminology) used by the system.

INERC features a pipeline of linguistic processors driven by a set of XSLT transformations which provide the control strategy to browse into the different document sections and separate the layout specific information from the textual content. The pipeline includes tokenization, terminological analysis, lexicon lookup, numeric entity recognition and ontology-driven entity classification. The Tokenizer transformation applies to the page content, segmenting the text into atomic tokens, classified as words, numbers and separators. The Terminology transformation recognizes terminological expressions as well as simple constituents and expresses them in their standardized form. Lexicon lookup matches lexical rules and entries against the input. This phase relies on the Italian lexicon and additional lexical tables for specific information (e.g. measurement units). A unit-matcher activates numerical expressions recognition in order to identify currencies, dates, lengths, and other domain specific quantities. Ontology lookup matches identified entities against the ontology and categorizes them accordingly.

To perform the above, INERC uses lexical resources, most of which are automatically generated from the domain ontology and the Italian lexicon. The INERC component focuses on the identification and classification of any relevant information related to the product to be identified. For example, in case the entity to be recognized is an operating system such as *Windows XP*, we want to recognize it in any surface linguistic expression: *Microsoft Windows XP Professional Edition*, *WinXP*, *Win XP*, etc. Finding the complete extension of a complex named entity could be relaxed if the partial information identified is sufficient to unambiguously detect a relevant concept in the text. Allowing such kind of "partial matching" of the information could make the system more robust with respect to new unforeseen surface representations and allows for easy tuning to new domains.

The customization method which has been implemented for INERC involves a statistically driven process of generalization from the annotated material to increase coverage of the observed (linguistic) phenomena. This requires, for each word that participates in a complex named entity, the evaluation of a frequency score to represent an estimated probability for that word being a useful marker for the NE category. This information is then used to automatically tune the lexical resources to new domains, building a set of (possibly partial) entries that bear strong semantic evidence of target categories.

## 5 RESULTS

### 5.1 Experimental Setting

We specified a common methodology for the collection of the necessary training and testing corpora for each domain and each language, which allows us to make a comparison of the results of the four language-specific NERC components. This methodology is comprised of two parts. First, we identify interesting characteristics of product descriptions (e.g. the preference for single or multi-product descriptions in a web page, information present in images, the amount of information present in a page) and collect relevant statistics from product descriptions for at least 50 different sites per language. In the second part of the collection process, we gather pages and create training and testing materials with a representative distribution according to the identified domain statistics.

The next stage is corpus annotation. We devised a common methodology and developed an annotation tool [self-reference, omitted]. The tool takes as input the ontology and the NERC DTD for the relevant domain and converts the names of the entities into a clickable menu. Annotations are stored as byte-offset files in the style of the Tipster[3] architecture but can also be merged directly into the XHTML document as XML elements. The corpus annotation methodology is comparable to standard annotation practice [2]. The annotation is based on a set of guidelines developed for the specific domain. Two human annotators use the guidelines in order to annotate the same pages. Then a 3rd person inspects the annotations produced and gives further instructions on the creation of the final annotations. Corpora of web pages for the two domains of the project were collected and annotated for each language using the above methodologies and the annotation tool. Table 1 provides the total number of named entities included in the

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

testing corpus along with the number of offerings (in parentheses) per language and domain.

**Table 1.** Laptop and Job offer corpora counts: entities (offerings).

| | *1st Domain* | *2nd Domain* |
|---|---|---|
| ***English*** | 5111 (423) | 842 (110) |
| ***French*** | 2400 (204) | 1738 (166) |
| ***Hellenic*** | 1759 (136) | 757 (128) |
| ***Italian*** | 3296 (267) | 1170 (156) |

### 5.2 Evaluation

In evaluating the mono-lingual NERC systems we follow the 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). The standard way to incorporate precision and recall into a single score is to compute *f-measure*. The IE community generally reports the harmonic mean which weights recall and precision equally (i.e. F = 2*(recall*precision)/(recall+precision). Table 2 shows f-measure scores across all categories of named entity for each of the four systems in the two domains of the project.

**Table 2.** Overall NERC Evaluation Results (f-measure)

| | *1st Domain* | *2nd Domain* |
|---|---|---|
| ***ENERC*** | 0.73 | 0.59 |
| ***FNERC*** | 0.77 | 0.75 |
| ***HNERC*** | 0.86 | 0.68 |
| ***INERC*** | 0.82 | 0.77 |

From a comparison of the NERC systems across the two domains, it is clear that the two systems continuing to use rule sets have maintained a greater degree of consistency in the face of sparser training material, while the two systems which have adopted machine learning for the 2nd domain have suffered a drop in performance (the number of entities and offerings in the second domain is significantly lower).

Additional evaluations performed for ENERC using cross-validation in order to maximize the amount of training data have demonstrated that performance of the ENERC machine learning system can be improved when more training data is available (overall f-measure increases from 0.59 to 0.67). The general conclusion to be drawn, then, is that the machine learning approaches are capable of performing as well as the rule-based ones *if enough training material is available*.

For the question of customization, the implications are that a certain amount of human labor cannot be avoided, either in fine-tuning rule sets or in annotating sufficient training material. We have come to the conclusion that the path to the swiftest customization is finding the best balance between these time investments. The level of expertise required for annotation is less than that required for tuning rules, though a high level of expertise is required for gathering representative corpora and developing annotation guidelines. Given the difficulties associated with gathering training material, an interactive approach (e.g. that adopted by FNERC, where machine learning assists the human developer of rule sets) appears to be very promising.

## 6 CONCLUDING REMARKS

The move from textual domains to web pages has been 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. The project partners have ported their NERC technologies to web pages taking into account the different document genres. Our approach compares favorably with other methods of information extraction from Web pages, such as 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 has considered adaptability a key design point and is rapidly extensible both to new languages and to new domains. Comparing the NERC systems across the two domains, the general conclusion to be drawn is that the machine learning approaches are capable of performing as well as the rule-based ones if enough domain-specific resources or training material is available. For the question of customization, given the difficulties associated with gathering training material, an interactive approach, where machine learning assists the human developer of rule sets, appears to be a very promising route to investigate.

## 7 REFERENCES

[1] E. Amitay. 1997. Hypertext: The importance of being different. MSc thesis, University of Edinburgh, September 1997.

[2] S. Boisen, M. Crystal, R. Schwartz, R. Stone and R. Wischedel "Annotating Resources for Information Extraction". LREC 2000. pp. 1211-1214. 2000.

[3] N. A. Chinchor. 1998. Overview of MUC-7/MET-2. http://www.muc.saic.com/ proceedings/muc_7_toc.html.

[4] J. R. Curran & S. Clark. 2003. Language Independent NER using a Maximum Entropy Tagger. CoNLL 2003.

[5] Proceedings of the 6th Computational Natural Language Learning Workshop (CoNLL-02), Taipei, Taiwan. http://cnts.uia.ac.be/conll2002/proceedings.html

[6] Proceedings of the 7th Computational Natural Language Learning Workshop (CoNLL-03), Edmonton, Canada. http://cnts.uia.ac.be/conll2003/proceedings.html

[7] D. Freitag, "Toward General-Purpose Learning for Information Extraction". COLING-ACL 1998.

[8] A. Laender, B. Ribeiro-Neto, A. da Silva, J. Teixeira A Brief Survey of Web Data Extraction Tools, ACM SIGMOD Records, vol. 31(2), June 2002.

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

[10] Quinlan, J. R., C4.5: Programs for machine learning, Morgan-Kaufmann, San Mateo, CA, 1993.