A Platform for Cross-lingual, Domain and User Adaptive Web Information Extraction

Vangelis Karkaletsis[1], **Constantine D. Spyropoulos**[1], **Claire Grover**[2], **Maria-Teresa Pazienza**[3], **Jose Coch**[4], **Dimitris Souflis**[5]

Abstract. This paper describes an advanced platform for web information extraction (IE) that enables customization to different domains, languages and users' interests. This platform was the result of the R&D project CROSSMARC which involved both academic and industrial organisations. The platform is composed of a core system for Web IE and a customization infrastructure. The system implements a distributed, multi-agent, open and multilingual architecture that integrates components for (a) collecting domain-specific web pages using crawling and spidering technologies, (b) extracting information from the collected web pages using natural language processing and machine learning techniques, and (c) presenting the extracted information according to users' interests employing user modelling techniques. The platform's customisation infrastructure provides an ontology management system and various customisation methods and tools for the creation of the application specific resources. The platform enables cross-lingual IE, supporting four languages in its current implementation, and has been tested in three different applications.

1 PROBLEM DESCRIPTION

The remarkable growth of the World Wide Web has led to an enormous increase in the amount and availability of on-line information. However, information is only valuable to the extent that it is accessible, easily retrieved and concerns the personal interests of the user. The growing volume of web data in various languages and formats, the lack of structured information, and the information diversity have made information and knowledge management a real challenge towards the effort to support the information society. It has been realized that added value is not gained merely through larger quantities of data, but through easier access to the required information at the right time and in the most suitable form. Enabling large scale information extraction (IE) from the Web is a crucial issue for the future of the Internet.

The traditional approach to Web IE is to create *wrappers*, i.e. sets of extraction rules, either manually or automatically. At run-time, wrappers extract information from unseen collections of Web pages, of known layout, and fill the slots of a predefined template. These collections are typically built by querying an appropriate search form in a Web site and collecting the response pages, which commonly share the same content format. The manual creation of wrappers presents many shortcomings due to the overhead in writing and maintaining them. On the other hand, the automatic creation of wrappers (wrapper induction – WI) presents also problems since a re-training of the wrappers is necessary when changes occur in the formatting of the targeted Web site or when pages from a "similar" Web site (i.e. under the same domain) are to be analysed. Training an effective site-independent wrapper is an attractive solution in terms of scalability, since any domain-specific page could be processed, without relying heavily on the hypertext structure.

The collection of the application specific web pages which will be processed by the wrappers (e.g. collecting web pages containing job offers descriptions from the web sites of IT companies) is a crucial issue. Thus, a collection mechanism is also necessary for the location of the application specific web sites (IT companies in the previous example) and the identification of interesting pages within them. The design and development of web pages collection and extraction systems needs to consider requirements such as: (a)enabling adaptation to new domains and languages, (b)facilitating maintenance for an existing domain, (c)providing strategies for effective site navigation, (d)ensuring personalised access, and (e)handling of structured, semi-structured or unstructured data.

The implementation of a web pages collection and extraction mechanism that addresses effectively the important issues mentioned above was the motivation for the development of a Web IE platform in the context of the R&D project CROSSMARC[6], which was partially funded by the EC. This paper describes the platform, presents one of the applications built using it and discusses the benefits and the open issues of the proposed approach for Web IE.

2 THE PLATFORM

CROSSMARC platform is composed of a core system for web information retrieval and extraction which can be trained to new applications and languages and a customization infrastructure that supports configuration of the system to new domains and languages. The core system implements a distributed, multi-agent, open and multi-lingual architecture which is depicted in Fig. 1. It involves components for the collection of interesting and domain-specific

[1] NCSR "Demokritos" (EL), Athens, {vangelis, costass}@iit.demokritos.gr
[2] University of Edinburgh (UK), grover@ed.ac.uk
[3] Universita di Roma "Tor Vergata" (IT), pazienza@info.uniroma2.it
[4] Lingway (FR), Paris, Jose.Coch@lingway.com
[5] Velti (EL), Athens, Dsouflis@velti.net

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

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.

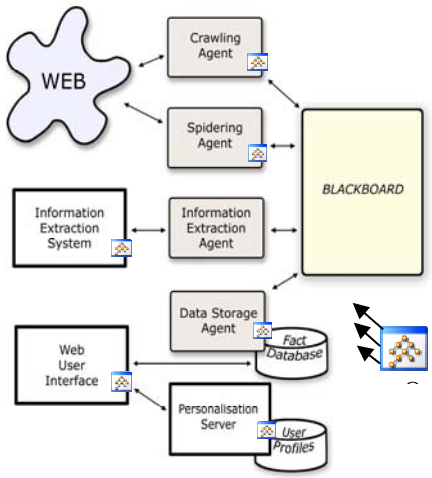


Fig. 1. System's agent based architecture

2.1 Customization Infrastructure

The infrastructure for configuring to new domains and languages involves: an ontology management system for the creation and maintenance of the ontology, the lexicons and other ontology-related resources; a methodology and a tool for the formation of corpus necessary for the training and testing of the modules in the spidering component; a methodology and a tool for the collection and annotation of corpus necessary for the training and testing of the information extraction components.

2.1.1 Ontology Management

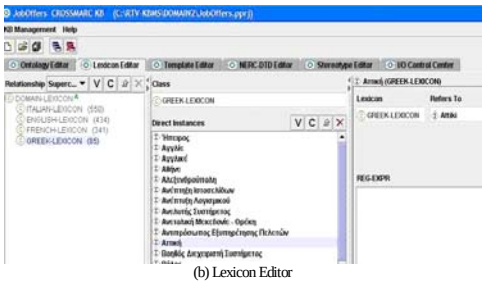
The ontology management system is based on the Protégé knowledge editor[7]. It provides a set of editors and functionalities [2] for: the creation and maintenance of domain ontologies (Fig. 2a); the creation and maintenance of lexicons under domain ontologies (Fig. 2b); the specification of the important entities for the domain (Fig. 2c); the specification of the important fact types for the domain, their relations to the NERC entities and their possible values according to the ontology (Fig. 2d); 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 the NERC DTD, the template as an XML schema, the stereotypes' definitions in XML.

2.1.2 Corpus Formation

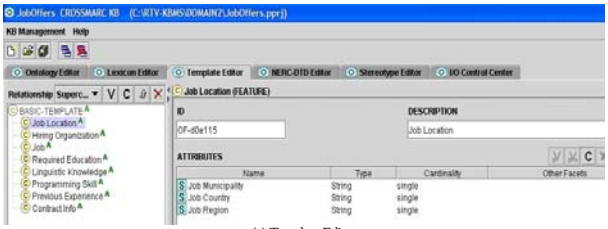
A simple approach was developed to facilitate the formation of training and testing corpus for the spidering component. This approach is based on an interactive process between the user (person responsible for corpus formation) and a simple machine learning based classifier, which is responsible for selecting Web pages and presenting them to the user for classification. It is realized by the *Corpus Formation Tool (CFT)*, which helps users build a corpus of positive and negative pages, with respect to a given domain. The steps of the corpus formation process are the following (see Fig. 3):

- The user downloads locally one or more Web sites ("Sites"), and selects the positive pages from them ("Pos").
- The *XMLVectorizer* module of the CFT is applied over the selected positive pages and the rest of downloaded pages taking as input the domain ontology and lexicon(s) and producing the vectorized pages ("pos.arff" and "unknown.arff" respectively). Then, the *Sampler* module of CFT takes these pages as input and selects from "Sites" the pages which are most similar to the positive pages ("Similar").
- Finally, the user classifies manually the "Similar" pages, moving the new positives into "Pos" and the negatives ones into "Neg".
- The above steps are repeated until we reach a pre-specified positive to negative pages ratio.

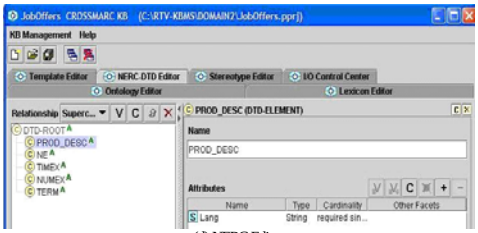
(a) Ontology Editor



(b) Lexicon Editor



(c) Template Editor



(d) NERC Editor

Fig. 2. Ontology Management System

[7] http://protege.stanford.edu/

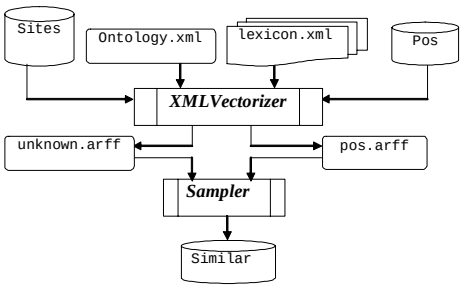


Fig. 3. Corpus Formation Process

2.1.3 Corpus Collection and Annotation

A corpus collection methodology was devised for the compilation of training and testing corpora that are representative and up-to-date. This methodology determines how different web pages characteristics must be taken into account and how they are to be represented in the corpora. According to this methodology, web pages in each domain are classified in categories (see Table 1). Category "A" pages contain a single offer, whereas "B" pages contain more than one. In "A1" pages the single offer description is not broken by irrelevant pieces of information, which is the case for "A2" pages. On the other hand, in "B1" pages, multiple offers descriptions (for jobs in this case) appear in different lines/rows of a page, in "B2" pages some of the descriptions may concern other types of offers, and in "B5" pages offer descriptions may appear in different columns of a table. The evaluation of the information extraction components takes into account this categorization providing not only total results but also results per page category in order to examine the effect of page characteristics to the system performance.

	English	French	Greek	Italian
TYPE A1	62%	47%	39%	34%
TYPE A2	0%	0%	0%	0%
TYPE A3	0%	0%	0%	0%
TYPE B1	34%	30%	55%	62%
TYPE B2	0%	0%	0%	0%
TYPE B3	0%	0%	0%	0%
TYPE B4	0%	0%	0%	0%
TYPE B5	4%	15%	6%	2%
TYPE B6	0%	0%	0%	0%
TYPE B7	0%	8%	0%	2%

Table 1. Pages categorization in the "job offers" domain

The creation of consistently annotated corpora is very important for the training and evaluation of IE systems. The corpus annotation methodology that has been developed is comparable to standard annotation practices for IE. The annotation task is based on guidelines that are issued for each new domain and on the use of an annotation tool [5]. Two human annotators annotate the same pages and a third one inspects their annotations and gives further instructions on the creation of the final annotations.

2.2 The System for Web Information Retrieval and Extraction

The system is distributed since the various language-specific IE modules can be found in different locations. This facilitates the integration of an IE module for a new language if this satisfies the I/O specifications of the architecture. Each language-specific IE module takes an XHTML page as input and returns the same page augmented with XML annotations marking the information found in the page, according to a common XML schema. The architecture is a multi-agent one since it involves agents for scheduling and supporting the project components. It is an open architecture since it provides clear I/O specifications for its major components facilitating the addition of new ones. Finally, it is multi-lingual since it enables the addition of new languages employing an ontology and the corresponding language-specific lexicons, adopting a common XHTML format for the output of the web pages collection sub-system, using a common DTD for the output of the named entity recognition & classification (NERC) processing stage of the IE sub-system, and using a common XML schema for storing the extracted information (this also allows the cross-lingual access to the extracted information).

The agent-based integrated prototype operates constantly according to the agent strategies and the initial settings of the administrator. The agents are configured by the administrator with the help of an XML configuration file which contains paths for ontologies, lexica and command files, various directories, network configuration constants, intervals and other numbers.

2.2.1 Web Pages Collection

Crawler implementation involves three different crawler versions [6]. The 1st one exploits the topic-based website hierarchies used by various search engines to return web sites under given points in these hierarchies. The 2nd one takes a given set of queries, exploiting the domain ontology and lexicons, submits them to a search engine, and then returns those sites that correspond to the pages returned. The 3rd one takes a set of 'seed' pages and then conducts a 'similar pages' search (available from advanced search engines such as Google) to find pages deemed similar to the seed pages. It then returns the sites corresponding to these pages. Each type of crawler can be adapted to different search engines or Web site hierarchies.

In focused crawling, the aim is to adapt the behaviour of the search engine(s) to the requirements of a user. On the other hand, in site-specific spidering, the spider navigates in a Web site, following best-scored-first links. Each Web page visited is evaluated, in order to decide whether it is really relevant to the topic, and its hyperlinks are scored in order to decide whether they are likely to lead to useful pages. Thus, a score-sorted queue of hyperlinks is constructed, which guides the retrieval of new pages. The spidering tool consists of three main modules: *site navigation, page filtering,* and *link scoring*. The whole process is depicted in Fig. 4.

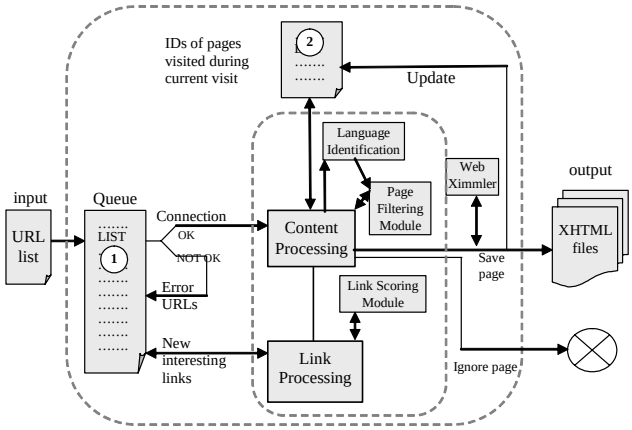


Fig. 4. Web Pages Collection

Before forwarding a page to "*Page Filtering*", this is processed by the "*Language Identification*" module which decides on the

language of the page using some simple rules that check for the occurrence of frequent words. According to its language, the page is forwarded to the corresponding filter. Before saving an interesting page this is transformed to XHTML using the Web Ximmler tool. The spider used in the final version of the prototype system integrates a machine learning based version of the page filtering module and a rule-based version of the link scoring module.

2.2.2 Information Extraction

The multi-lingual Information Extraction (IE) system integrates four mono-lingual IE sub-systems which operate as autonomous processors that can be found in different locations (see Fig. 5). For interfacing with the IE sub-systems a proxy mechanism was developed (IERI) which takes the XHTML pages produced by the spider and routes them to the corresponding monolingual sub-system according to their language. Each IE sub-system relies on the following components: *named entity recognition (NERC)*, *demarcation* and *fact extraction (FE)*.

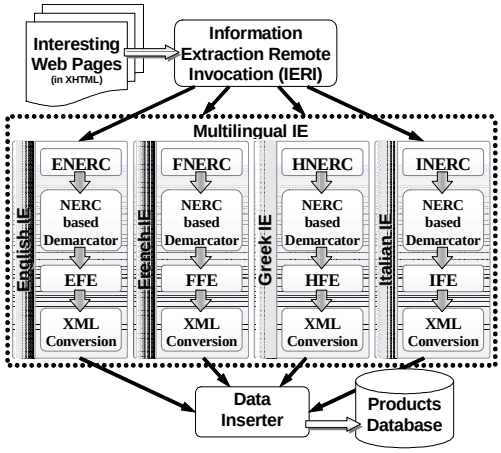


Fig. 5. Architecture of the multi-lingual IE system

NERC identifies domain-specific named entities in pages from different sites [1]. All the initial versions of the monolingual NERC modules were rule-based ones sharing similar architectures. Each NERC system uses the appropriate lexicon which has to be maintained by adding synonym variants of the basic entries. During the last stage of the project, the teams developing English and Hellenic NERC (ENERC and HNERC in Fig. 4) switched to machine learning systems, while the Italian system (INERC) remained rule-based. The French system (FNERC) followed a hybrid approach using machine learning techniques to induce human editable NERC rules.

The output of the monolingual NERC systems is processed by the Demarcation component, which has to identify how many different product/offer descriptions exist within an XHTML page and classify the recognized entities in the corresponding products/offers. Demarcation is a complex task, which becomes even more complicated by the visual aspect of web pages. We developed a heuristics-based demarcator for both domains of the project. The optimal set of heuristics for the new domain is formalized, through a series of experiments.

FE identifies domain-specific facts, i.e. assigns domain-specific roles to entities. Four different FE approaches were implemented sharing the same basic idea, that of exploiting the linguistic results of NERC in order to produce site independent domain-specific IE systems, thus handling the main problem of the existing wrapper induction techniques. Two of the FE modules re-implement well-established wrapper induction technique, one treats FE as a classification task using machine learning, and one follows a hybrid approach where the model learned by a machine learning algorithm is in a format that can be edited by knowledge engineers [7].

The extracted facts are stored in the products' database, normalising them first according to the ontology. This is necessary in order to present them to the end-user according to the user's preferred locale. The extracted information is then converted into a common XML representation, which is used by the Data Storage agent to feed the products database.

2.2.3 Data Storage and Presentation

The data storage component stores the extracted facts, from the XML file produced by IE, into domain-specific databases according to the following principles: (a)all facts belonging in the same product/offer form a record in the database, (b)in each record the page URL and the storage date-time are also kept, (c)all records extracted from the same web page form a dataset.

The UI was implemented as an internationalised application in order to facilitate the porting to new languages and domains. The extracted information is presented according to the preferences of the end-user exploiting the functionalities of a general-purpose personalization server which is integrated with the UI.

3 APPLICATION BUILDING

So far, the platform has been used for the development of three applications to extract information from:

- laptops offers in e-retailers web sites (in four languages),
- job offers in IT companies web sites (in four languages),
- holidays' packages in the sites of travel agencies (in two languages).

The building of an application involves two main stages. The 1st one concerns the creation of the application-specific resources using the customization infrastructure whereas the 2nd phase concerns the training of the integrated system using the application-specific resources. For example, building the application for the domain of "job offers", it involves the following steps for the 1st stage.

After studying the domain, important concepts such as *"Location", "Employer Organisation", "Job Title",* etc., concept-subconcept relationships (e.g. "*Country*" is-a "*Location*"), and concepts instances (e.g. "*Attiki*" is an instance of "*Greek Region*") are created using the ontology editor (see Fig. 2a). The linguistic realizations, in each language, for the concepts names and instances are then created using the lexicon editor (see Fig. 2b). It is possible to add more than one realization (synonyms). The instance "*Greece*" for example is realized by «*Ελλάδα*» and «*Ελλάς*» in the Greek lexicon.

The types of important entities inside offers descriptions along with their attributes are specified using the NERC editor (see Fig. 2c). This enables the forming of a common NERC DTD for all monolingual NERC systems. In the case of "job offers", there are named entities (NE), numeric expressions (NUMEX), time expressions (TIMEX) and terms. These are children of the offer description "PROD_DESC" which has "Lang" as an attribute (this stands for the language of the web page). The important features for the domain and their attributes are then specified using the template editor (se Fig. 2d). These attributes take values from the ontology. For instance, the "*Job Location*" feature has as attributes "*Job Country*", "*Job Region*" and "*Job Municipality*" which take as values the instances of the corresponding concepts in the ontology.

These attributes will form the FE schema for the domain which is exported by the template editor.

The corpus for the spidering component is collected next using the corpus formation tool, whereas the corpus for the monolingual IE sub-systems is collected using the corpus collection methodology and the annotation tool. Details on both corpora (number of positive and negative pages per language and application for the first one, number of pages, offers and entities per language and application for the second one) can be found in [3]. Finally, the features of each user stereotype are specified using the stereotypes editor.

During the 2nd stage, the 1st step is to configure the crawler. This is done through a series of experiments with different parameters for the three crawler versions as well as with different combinations of them. The evaluation results for the best combination can be found in [6]. The page filtering and the link scoring modules of the spider are trained and tested using the collected corpus, and the trained modules are then integrated in the spider. The corresponding evaluation results can be found in [3]. Each monolingual IE sub-system is then trained using the collected corpus. Training involves the manual writing of rules (Italian IE), or the use of machine learning (English, Greek IE), or even the use of a hybrid approach (French IE). The evaluation results for the 2nd domain presented in Table 2 (for both domains see in [3]).

	F-measure (%)
English IE	50,7
French IE	66,0
Greek IE	53,5
Italian IE	67,8

Table 2. IE results for the 2nd domain

The final step concerns the customization of the UI exploiting the ontology and the corresponding lexicons as well as the stereotypes definitions. Some language-specific parts of the UI that are not contained in the ontology are translated manually.

4 BENEFITS, OPEN ISSUES

CROSSMARC platform employs most of the categories of web extraction tools presented in [4]. It uses:

- Wrapper Induction (WI) techniques for the fact extraction stage in order to exploit the formatting features of the web pages.
- NLP techniques to exploit linguistic features of the web pages enabling the process of domain specific web pages in different sites and in different languages (multilingual, site-independent).
- Ontology engineering to enable the creation and maintenance of ontologies, language-specific lexica as well as other application-specific resources.

The CROSSMARC integrated system and its components were evaluated during the building of the applications. The main conclusions from these evaluations are summarized below.

The crawler evaluation measured how well the system performs in finding 'fit' sites. In our experiments, acceptable performance was achieved by putting increased effort into the initial stage of forming hypotheses about what would be good directory and query start points [6]. The main conclusion of spider evaluation is that we are able to identify with a fairly high degree of confidence, when a Web page is an interesting one according to the application.

The level of performance achieved by all IE sub-systems is satisfactory, especially for offer descriptions extracted from simpler web pages (A1, B1 pages) which represent the large percentage in the domains examined in CROSSMARC. In addition, the existing sub-systems can be tuned further in order to achieve better performance. In terms of NERC, the main conclusion 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. Although demarcation presents good performance in web pages of simpler structure, the results are not good enough in more complex pages. This affects the performance of the whole IE system. Concerning FE, the approaches based on wrapper induction and classification give similar and satisfactory results showing that the exploitation of linguistic output can improve significantly the performance of existing techniques without requiring training for each web site separately.

The first two applications are accessible from CROSSMARC site. The visitors can also evaluate the performance of the system components as well as the entire system using a web-based evaluation questionnaire following specific evaluation scenarios.

5 CONCLUDING REMARKS

We have described an operational platform for information retrieval and extraction from web pages which provides a trainable system and a customization infrastructure. The system implements a distributed, multi-agent, open and multilingual architecture integrating components based on state of the art AI technologies and commercial tools. In the future, more advanced components will be tested and integrated in the platform. In the meantime, the system will be accessible from CROSSMARC site giving the ability to interested visitors to use and evaluate it. In addition, the various resources and corpora for the two first applications will be available shortly from the site for research purposes.

6 REFERENCES

[1] Grover C., S.McDonald, V.Karkaletsis, D.Farmakiotou, G.Samaritakis, G.Petasis, M.T.Pazienza, M.Vindigni, F.Vichot and F.Wolinski. "Multilingual XML-Based Named Entity Recognition", Proceedings of LREC-2002, Las Palmas, Spain, May 2002.

[2] Hachey B., C. Grover, V. Karkaletsis, A. Valarakos, M.T. Pazienza, M. Vindigni, E. Cartier, J. Coch. "Use of Ontologies for Cross-lingual Information Management in the Web", Proceedings of the Ontologies and Information Extraction International Workshop, EUROLAN 2003, Bucharest, Romania, July 28 - August 8, 2003.

[3] Karkaletsis V. and C.D. Spyropoulos, "Information Retrieval and Extraction from the Web: the CROSSMARC approach", Proceedings of the RIAO 2004 Conference, Toulouse, France, April 2004.

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

[5] Sigletos G., D. Farmakiotou, K. Stamatakis, G. Paliouras, V. Karkaletsis. "Annotating Web pages for the needs of Web Information Extraction applications", Poster at WWW-2203 Conference, Budapest, Hungary, May 20-24, 2003.

[6] Stamatakis K., V. Karkaletsis, G. Paliouras, J. Horlock, C. Grover, J.R. Curran, S. Dingare. "Domain-Specific Web Site Identification: The CROSSMARC Focused Web Crawler", Proceedings of WDA 2003, Edinburgh, UK, August 3, 2003.

[7] CROSSMARC Public Final Report . http://www.iit.demokritos.gr/skel/crossmarc/AnnualReport2003